НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

«На правах рукопису»

УДК _____

«До захисту допущено»
В.о. завідувача кафедри
_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)
“___”_____2019 р.

Магістерська дисертація

121 «Інженерія програмного забезпечення»
зі спеціальності
на тему: «*Інтелектуальна система дослідження та аналізу текстів* »

Виконав : студент VI курсу, групи *ІП-82мп*
Зубрицький Андрій Юрійович
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *доц., доц., к.т.н. Фіногенов О.Д,*
(посада, науковий ступінь, , вчене звання, прізвище та ініціали) (підпис)

Консультант *доц., к.т.н., Ліщук К.І.*
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент доц., доц., к.т.н. Яблонський П.М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Факультет *інформатики та обчислювальної техніки*
(повна назва)

Кафедра *автоматизованих систем обробки інформації та управління*
(повна назва)

Рівень вищої освіти *другий (магістерський) за освітньо-професійною програмою*

Спеціальність *121 «Інженерія програмного забезпечення»*
(код і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ *О.А. Павлов*

«___» _____2019 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Зубрицькому Андрію Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації *Інтелектуальна система дослідження та аналізу текстів*

науковий керівник дисертації *к.т.н., доц. Фіногенов О.Д.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "____" _____ 20__ р. № _____

2. Строк подання студентом дисертації "____" _____ 20_ р.

3. Об'єкт дослідження *Методи дослідження тексту, серед яких морфологічний аналіз, частотний аналіз та розпізнавання тональності тексту*

4. Предмет дослідження *Методи наївного байєсівського класифікатора та частотного аналізу*

5. Перелік завдань, які потрібно розробити *Проаналізувати різні системи для дослідження; проаналізувати різні методи; спроектувати архітектуру для системи; розробити систему, яка реалізовує запропоновані методи проаналізувати ефективність використаних методів*

6. Перелік графічного матеріалу
Функціональна схема

Схема

7. Орієнтовний перелік публікацій *Аналіз систем для дослідження тексту лінгвістами. Проектування архітектури системи дослідження тексту*

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання “ *01* ” *вересня* 20 *19* р

Календарний план

	№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
	1	*Підготовка та оформлення вступу*	*15.09.2019*	
	2	*Порівняльний аналіз існуючих систем дослідження тексту*	*2.10.2019*	
	3	*Постановка та реалізація обраних методів дослідження тексту*	*12.10.2019*	
	4	*Проектування архітектури системи*	*28.10.2019*	
	5	*Розробка програмного забезпечення*	*5.11.2019*	
	7	*Проведення експериментальних досліджень реалізованих методів*	*14.11.2019*	
	8	*Оформлення документації*	*27.11.2019*	
	9	*Подання роботи на попередній захист*	*05.12.2019*	
	10	*Подання роботи на основний захист*	*16.12.2019*	

Студент _____ (підпис) _____ (ініціали, прізвище)

Науковий керівник _____ (підпис) _____ (ініціали, прізвище)

РЕФЕРАТ

Актуальність теми: відсутність систем для дослідження тексту українською мовою

Мета дослідження: створення математичного та програмного забезпечення для аналізу українського тексту для дослідників-лінгвістів

Для реалізації поставленої мети були сформульовані **наступні завдання**:

- проаналізувати існуючі системи для дослідження тексту
- проаналізувати існуючі методи дослідження тексту
- реалізувати три методи дослідження тексту
- спроектувати архітектуру системи, що масштабується
- розробити веб-додаток для використання системи
- проаналізувати ефективність реалізованих методів
- розробити стартап-проект для створеної системи

Об’єкт дослідження: класи методів аналізу тексту

Предмет дослідження: аналіз українського тексту на різних рівнях мови

Методи дослідження: теоретичні, емпіричні, порівняння, аналіз

Наукова новизна: Найбільш суттєвими науковими результатами магістерської дисертації є:

- створено модель розпізнавання тональності українського тексту
- запропоновано архітектура системи, що масштабується та містить можливість додавання модулів до неї

Практичне значення отриманих результатів визначається тим, що реалізовану систему дослідники лінгвісти можуть використовувати для аналізу тексту

Зв'язок роботи з науковими програмами, планами, темами: наукові дослідження лінгвістів

Апробація: Основні положення роботи доповідались і обговорювались на двух конференціях

Публікації: Наукові положення дисертації опубліковані в двух наукових журналах.

Ключові слова: аналіз тексту, комп'ютерна лінгвістика, лінгвістичні системи, обробка природної мови, морфологічний аналіз, частотний аналіз, аналіз тональності тексту

ABSTRACT

The relevance of the topic: the absense of a system for text study in Ukrainian

The purpose of the study: Creating systems for the research of Ukrainian text by linguists

The following tasks were set:

- analyze different systems for text research
- analyze different methods for text research
- implement three methods of text reasearch
- design the scalable architecture for the system
- develop a web application for using of the system
- analyze the effectiveness of the implemented methods
- to develop a startup project for the created system

The object of research: methods of the text research, including morphological analysis, statistical analysis and sentiment analysis of the text using machine learning

The subject of research: methods of naive Bayesian classifier and most often.

Methods of research: theoretical, empirical, comparison, analysis

Originality of research: The most significant scientific results of research are:

- model of tone recognition of Ukrainian text was created
- an architectural system is proposed that scales and adds modules that are added to each

The practical meaning of the results is that linguists can use the system for a text research

Connection with scientific programs, plans, topics: scientific researches of linguists

Testing: The main functions of the paper were reported and discussed at two conferences

Publications: The scientific works of theses are published in two scientific journals.

Keywords: text analysis, computer linguistics, linguistic systems, natural language processing, morphological analysis, statistical analysis, sentiment analysis

ЗМІСТ

ВСТУП

На даний момент велика кількість дослідників лінгвістів використовують застарілі методи дослідження тексту, через що витрачають багато свого часу та ресурсів. Для того, щоб виконувати свою роботу більш ефективно їм потрібно використовувати інформаційні технології для виконання дослідження тексту. При цьому дослідники можуть використовувати певні програми, які можуть аналізувати український текст певним чином, але не вистачає саме комплексного рішення для виконання дослідження. Наявність такої системи для аналізу тексту могла б значно полегшити життя дослідників-лінгвістів та автоматизувати їх роботу.

При цьому існують деякі системи для дослідження тексту, але в основному ці системи можуть використовуватися тільки для англійської мови. При цьому у ході виконання порівняльного аналізу існуючих систем для українського мови наявного рішення не було виявлено.

Отже існує потреба у реалізації системи, яка створить робоче місце лінгвіста, який за допомогою браузера може зайти в цю систему, авторизуватися, додати текст та почати його аналіз за допомогою різних методів. При цьому потрібно розробити таку архітектуру системи, яка дозволить її легко масштабувати та розширювати функціонал для нових методів аналізу.

Таким чином метою роботи є розробка система для дослідження та аналізу українського тексту лінгвістами, реалізація трьох видів аналізу, серед який морфологічний аналіз, статистичний аналіз та аналіз тональності тексту за допомогою машинного навчання.

Для досягнення поставленої мети були вирішені наступні задачі:

- проаналізовано існуючі системи для дослідження тексту
- проаналізовано існуючі методи дослідження тексту

- реалізовано три методи дослідження тексту
- спроектовано архітектуру системи, що масштабується
- розроблено веб-додаток для використання системи
- проаналізовано ефективність реалізованих методів
- розроблено стартап-проект для створеної системи

Об’єктом дослідження є методи аналізу тексту, серед яких морфологічний аналіз, статистичний аналіз та аналіз тональності тексту за допомогою машинного навчання.

Предметом дослідження є методи наївного байєсовського класифікатора та частотного аналізу.

Наукова новизна роботи полягає у наступному:

- проаналізовано існуючі системи для дослідження тексту, зокрема українського
- проаналізовано основні методи дослідження тексту
- створено модель розпізнавання тональності українського тексту
- створено систему для дослідження українського тексту лінгвістами

Апробація роботи. За результатами даної роботи була підготовлена дві статті. Перша стаття з темою “Аналіз систем для дослідження тексту дослідниками-лінгвістами” була опублікована у журналі “”. Друга стаття з темою “Проектування архітектури системи дослідження та аналізу тексту” була опублікована в журналі “”.

У ході виконання роботи було реалізовано три методи дослідження тексту, серед яких морфологічнний аналіз, частотний аналіз та аналіз тональності тексту за допомогою машинного навчання.

Практичними результатами роботи є реалізація веб-додатку для дослідження та аналізу тексту українською мовою.

Робота містить 4 розділи. В першому розділі виконується аналіз предметної області. У другому розділі виконується опис методів дослідження тексту, реалізованих у роботі. У третьому розділі виконується опис програмного забезпечення, що реалізовано у роботі, описано архітектурне рішення, основні його компоненти та сутності. Четвертий розділ присвячений розробці стартап-проекту на основі даної магістерської роботи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблема аналізу тексту

Аналіз тексту - це галузь інформатики та машинного навчання, що стосується взаємодії між комп'ютерами та людськими мовами. У сучасному світі аналіз тексту використовується для розуміння як текстових так і голосових повідомлень.

Наприклад, ми можемо використовувати аналіз тексту для створення систем, що розпізнають мовлення, узагальнюють документи, роблять машинний переклад, виявляють спам, розпізнають назви, відповідають на запитання, роблять автодоповнення тексту, передбачають текст, що користувач буде вводити.

NLP - галузь штучного інтелекту, яка допомагає комп’ютерам зрозуміти людську мову. NLP спирається на багато дисциплін, включаючи інформатику та обчислювальну лінгвістику, намагаючись заповнити прогалину між людським спілкуванням та комп'ютерним розумінням.

Хоча обробка природних мов не є новою наукою, технологія швидко розвивається завдяки підвищеному інтересу до комунікацій між людиною та машиною, а також наявності великих даних, потужних обчислень та вдосконалених алгоритмів.

Як людина, ви можете говорити і писати англійською, іспанською або китайською мовами. Але рідна мова комп'ютера відома як машинний код або машинна мова для багатьох людей незрозуміла.

Програмісти використовували перфокарти для спілкування з першими комп’ютерами 70 років тому. Цей ручний і важкий процес зрозуміла порівняно невелика кількість людей. Тепер можливо сказати: "Алекса, мені подобається ця пісня", і пристрій, який відтворює музику у вашому домі, зменшить гучність і відповість: "Добре. Рейтинг збережений ", людським голосом.

Сьогодні більшість із нас мають смартфони, які розпізнають мовлення. Ці смартфони використовують аналіз тексту, щоб зрозуміти, про що говориться. Також багато людей використовують ноутбуки, операційна система яких має вбудоване розпізнавання мови. Приклади таким систем можуть бути дві найбільш відомі: Cortana та Siri. Cortana - це голосовий асистент від Microsoft, що дозволяє за допомогою голосу налаштовувати нагадування, відкривати додатки, надсилати повідомлення на електронну пошту, грати в ігри, відстежувати рейси, перевіряти погоду тощо. За допомогою другого відомого додатку Siri користувач може зателефонувати, надіслати повідомлення кому-небудь, надіслати електронний лист, встановити таймер, сфотографувати, відкрити додаток, встановити будильник, користуватися навігацією тощо.

Відомий веб додаток для обміну електронними листами Gmail також використовує аналіз тексту для виявлення спаму у повідомленнях.

Аналіз тексту включає ряд етапів:

- графематичний аналіз: виділення кордонів слів, речень, абзаців та інших елементів тексту (наприклад, врізок в газетному тексті);

- морфологічний аналіз: визначення початкової форми кожного використаного в тексті слова і набору морфологічних характеристик цього слова;

- синтаксичний аналіз: виявлення граматичної структури речень тексту;

- семантичний аналіз: визначення сенсу фраз.

Графематичний аналіз визначається також як токенізація (від англ. token - окреме слово, фраза або будь-який інший елемент тексту). Формальними сигналами межі текстових елементів виступають роздільники різного роду: пробіл, позначає межу між словами, великі літери та розділові знаки, що

позначають межі між реченнями і складовими частинами речень, абзацний відступ, що позначає межі між зв'язаними за змістом групами речень тощо .

Однак формальний метод визначення меж слів можна застосувати не завжди. Наприклад, в китайській мові немає формальних кордонів слів. Крім того, навіть в поширених європейських мовах існують стійкі поєднання слів, розділені пропуском, які слід сприймати як одну лексему, наприклад, New York. Очевидно, що такі випадки слід враховувати в системах графематичного аналізу, наприклад, шляхом створення списків багатослівних лексем.

При морфологічному аналізі кожне використане в тексті слово зводиться до його початкової форми і визначається набір морфологічних характеристик текстової форми слова: частина мови; рід, число і відмінок для іменників, число, особа для дієслів і т.п.

Кожне вжите в тексті слово називається словоформою (або слововживанням). Для забезпечення зв'язності тексту потрібне повторення тих же самих слів, тому нерідко різні словоформи одного або декількох речень тексту зводяться до однієї і тієї ж вихідної форми.

При синтаксичному аналізі необхідно визначити роль слів у реченні і їх зв'язок між собою. Результатом цього етапу автоматичного аналізу є уявлення синтаксичних зв'язків кожного речення у вигляді моделей, наприклад у вигляді дерева залежностей.

Проблемою синтаксичного аналізу виступає наявність альтернативних варіантів синтаксичного розбору (синтаксичної багатозначності). Виникнення синтаксичної багатозначності обумовлюється лексико-морфологічною багатозначністю словоформ (одна і та ж словоформа може відноситися до різних вихідним форм або до різних морфологічних форм однієї лексеми), а також неоднозначністю самих правил розбору, які можуть мати на меті представлення синтаксичної структури, наприклад, у вигляді дерева залежностей.

Семантичний аналіз являє собою, мабуть, найбільш складний напрямок автоматичного аналізу тексту. У цьому випадку потрібне встановлення семантичних відносин між словами в тексті, об'єднання різних мовних виразів, що відносяться до одного і того ж поняття.

1.2 Підготовка тексту до аналізу

Для підготовки тексту до аналізу найчастіше за все використовують такі методи як:

- Поділ тексту на речення
- Поділ тексту на слова
- Зведення слів до їх початкової форми
- Видалення неважливих для аналізу слів
- Bag of words
- TF-IDF

В багатьох мовах ми можемо розділяти речення, коли бачимо розділовий знак. Однак ця проблема не є тривіальною, так як наприклад точка часта викорстовується також для абревіатур. В такому випадку нам може допомогти словник зі скороченнями, що дозволить уникнути неправильного поділу на речення. На даний час існує достатня кількість бібліотек, яка містить в собі словники та дозволяє виконувати правильний поділ тексту на речення.

Токенізація слів - це проблема поділу тексту складові слова. В багатьох мовах пробіл є гарним наближенням розділювача слів.

Однак у нас все ще можуть виникнути проблеми, якщо ми лише використовуємо пробіл для поділу на слова. Наприклад існують поняття, що складаються з декількох слів. Тому у даному випадку ми також часто використовуємо словник для поділ тексту на слова. Зараз дуже часто використовуються готові бібліотеки для досягнення бажаних результатів, які вже містять використання словників у своїй реалізації.

Документи дуже часто можуть містити різні форми слова, такі як викладати, викладав, викладання. Також іноді ми маємо споріднені слова з подібним значенням, наприклад, нація, національний, національність.

Мета зведення слів до їх початкової форми - звести похідні споріднені форми слова до загальної основної форми для того щоб аналіз тексту був більш продуктивним.

Є два способи зведення слів до їх початкової форми. Перший просто обрізає закінчення слів, при цьому не беручи до уваги морфологічне значення слова.

Другий спосіб полягає у тому, що зведення виконується за допомогою словника та морфологічного аналізу слів, як правило, метою є видалення закінчень і при цьому повернення до основної чи словникової форми слова, яка відома як лема.

Різниця полягає в тому, що перший спосіб працює без знання контексту, і тому не може зрозуміти різницю між словами, які мають різний зміст залежно від місця в речення. Але у першого способу є також деякі переваги, його легше реалізувати і зазвичай він відбувається швидше. Крім того, для деяких застосувань знижена "точність" може не мати значення.

Зайві слова можуть заважати продуктивному аналізу тексту так як дуже часто вони не мають важливого значення для аналізу. Ось чому зазвичай ці невідповідні слова видаляють.

Такими словами зазвичай є найпоширеніші слова, такі наприклад як артиклі "і", "the", "a" у англійській мові, але не існує єдиного універсального списку таких слів. Список таких слів може змінюватися залежно від вашого додатку.

1.3 Аналіз існуючих рішень

На даний момент існує дефіцит систем для дослідження тексту українською мовою, до того ж велика кількість дослідників лінгвістів

використовують застарілі методи дослідження тексту, через що витрачають багато свого часу та ресурсів. Для автоматизації їх роботи вони потребують наявності системи для аналізу тексту. За мету даного аналізу було поставлено дослідження існуючих систем аналізу тексту для лінгвістів, їх функціональності та аналіз систем, що працюють з українською мовою. Таким чином за допомогою даного аналізу ми зможемо зрозуміти які системи існують на даний момент для українських дослідників та сформувати основний набір стандартних функцій.

Перша група комп'ютерних систем призначена для синтаксичного і морфологічного аналізу текстів. Інформація про граматику є однією з найважливіших при формуванні цілісного уявлення про систему мови, так що ці програми можуть бути корисні в нашому дослідженні

Друга група систем автоматичної обробки текстів об'єднує продукти, які дозволяють прийти до узагальненого подання про частоту виявлених лексичних одиниць, їх групування у текстах, а також дають підстави для дослідження семантичних процесів в досліджуваних мовних продуктах.

До третьої групи систем належать системи аналізу тексту з використанням методів машинного навчання. Такі системи навчаються на великій кількості даних та розпізнають текст на основі даних з датасетів. Основними задачами, які вирішують такі програми є класифікація текстів, вибірка слів та визначення теми тексту.

Для того щоб користуватися системою для дослідження тексту потрібно в неї якийсь текст додати. Після того як текст буде додано над ним можна виконувати різноманітні операції та виконувати його аналіз. За типом введення можна виділити:

- системи із звуковим введенням
- системи із розпізнаванням тексту з друкованих носіїв

- системи з введенням за допомогою клавіатури

У системах із звуковим введенням додавання тексту відбувається за допомогою мікрофону, після чого система виділяє у звуковому потоці окремі знаки та намагається розпізнати знаки мови.

Створення систем аналізу мови, які виконують перетворення інформації із звукового формату у текстовий передбачає співпрацю представників самих різних спеціальностей. Для економії часу і зусиль вчених і практиків різні компанії, в тому числі Microsoft, випускають готові програмні модулі й інтерфейси, що перетворюють мовлення у текст. Завдяки цьому є можливість використовувати ці модулі у власних розробках. Правда, в цьому випадку мовні можливості системи, на жаль, обмежені використаними засобами та технологіями. Наприклад, велика кількість таким систем не працює з українською мовою.

Системи із розпізнаванням тексту з друкованих носіїв частіше за все отримують текст зі сканеру та розпізнають текст за допомогою автоматичного розпізнавання символів у зображеннях. Знову ж реалізувати такі системи з нуля достатньо складно, але є достатня кількість існуючих модулів, які можна використати у своїй системі.

Систем із веденням за допомогою клавіатури серед проаналізованих існує найбільша кількість, так як їх легше всього реалізувати.

Серед багатьох систем із використанням машинного навчання, які було розглянуто, можна виділити MonkeyLearn.

Вона дозволяє використовувати класифікацію текстів, виділення тем, аналіз настроїв у тексті, вибірку окремих слів. При цьому система натренована на певних датасетах, наприклад відгуки фільмів. Тому для використання моделей потрібно використовувати тексти, що відповідають тематиці даних.

Серед недоліків можна виділити те, що система є комерційною, а також більше призначена для вирішення бізнес задач, ніж для дослідження тексту.

Серед переваг можна виділити те, що у системі реалізована велика кількість способів аналізу тексту за допомогою методів машинного навчання. А також вона має гарний користувацький інтерфейс, що реалізований у веб-браузері і дозволяє використовувати систему без її попередньої інсталяції.

Наступною системою, яка виконує аналіз за допомогою статистики є TextAnalyst 2.0. Вона є інструментом аналізу символьних текстів. Дозволяє побудувати семантичну мережу понять, виділених в оброблюваному тексті, з посиланнями на контекст. Окрім цього є можливість пошуку по змісту у фрагментах тексту з урахуванням прихованих в тексті змістових зв'язків. Дозволяє аналізувати текст шляхом побудови ієрархічного дерева тем / підтем, яких торкається в тексті. Також є можливість реферування тексту.

Крім окремого продукту TextAnalyst, також пропонується інструментарій розробника TextAnalyst SDK, що включає функції лематизації (приведення слів до нормальної форми) для російської та англійської мов, побудови частотних списків понять, пошуку слів в контексті і т.д.

Ще одна компонента, TextAnalyst Lib, може використовуватися для побудови гіпертекстових електронних книг.

Всі компоненти реалізовані для Windows.і доступні для безкоштовного завантаження.

Основні можливості:

- аналіз змісту тексту з автоматичним формуванням семантичної мережі з гіперпосиланнями

- виявлення семантичної структури тексту у вигляді ієрархії тем і підтем;

- розумовий пошук з урахуванням прихованих смислових зв'язків слів запиту зі словами тексту;

- автоматичне реферування тексту - формування його змістового портрету в термінах найбільш інформативних фраз;

- кластеризація інформації - аналіз розподілу матеріалу текстів по тематичним класів;

- автоматичне формування повнотекстової бази знань з гіпертекстової структурою і можливостями асоціативного доступу до інформації.

Серед переваг системи можна виділити те, що вона має велику кількість реалізованого функціоналу та може використовуватися безкоштовного.

Серед недоліків можна виділити те, що немає веб-версії системи, а також не існує можливості працювати з українською мовою.

Наступною системою, що виконує аналіз тексту за допомогою статистики є NetXtract. Ця система дозволяє швидко отримати впорядкований індекс слів в завантаженому HTML документі. Індекс може бути впорядкований за алфавітом або частотою. Для кожного слова в індексі можна досліджувати контекст, в якому це слово зустрічається.

Вибрані слова за бажанням заносяться в персональну базу знань, яка дозволяє систематизувати знайдені документи зручним чином. За допомогою цієї програми можна швидко знайти потрібну інформацію на веб-сторінках і в документах, а потім зберегти її у власній базі даних. NetXtract автоматично індексує кожен документ, який відображається, виділяє всі контексти для будь-якого ключового терміну за вибором і дозволяє вибирати найбільш цікавий контекст. Таким чином програма призначена для систематизації лексичних одиниць і виявлення закономірностей їх використання та для розуміння середовища формування нових лексичних значень використання цієї програми.

Розглянемо наступну систему, що виконує аналіз тексту за допомогою статистичних засобів - WordStat. Вона пропонує підрахунок частоти виникнення різних слів в текстових або html-файлах. Розуміє основні російські кодування, ігнорує html розмітку. WordStat може бути використаний для швидкого вилучення та аналізу інформації з великої кількості документів.

Використовується для:

- контент-аналізу: відкриті відповіді, інтерв'ю або фокус-групи, стенограми;
- бізнес-аналітики і конкурентного аналізу веб-сайтів;
- вилучення інформації і знань зі звітів про інциденти, скарг клієнтів;
- контент-аналізу новин або наукової літератури;
- автоматичного маркування та класифікації документів;
- виявлення випадків шахрайства, авторства, патентного аналізу.
- є найбільш використовуваним сервісом, який показує статистику ключових слів і допомагає в прогнозуванні трафіку.

Серед переваг системи можна виділити те, що вона дозволяє аналізувати веб-сторінки та показує велику кількість статистичних даних.

Серед недоліків можна виділити те, що для роботи із системою потрібні мати певні знання зі статистики, а також те, що вона не підтримує українську мову.

Розглянемо систему, яка містить у собі лінгвістичні способи аналізу тексту WordTabulator. Вона дозволяє аналізувати тексти в середовищі Windows. Має вбудований морфологічний модуль, що дозволяє шукати всі видозміни російських слів, заданих базовою формою. Дозволяє виконувати контекстний перегляд результатів, представлених у вигляді гіпертекстового індексу та

містить можливість аналізу двох текстових корпусів на схожість або відмінність.

Далі було проаналізовано систему, яка містить у собі поєднання різних видів аналізу - Langsoft. Вона дозволяє виконувати різні види обробки природної мови (англійської та німецької):

- граматичний розбір речень;
- перевірка орфографії і граматики;
- інтелектуальне перетворення тексту (автоматична редакторська правка);
- резюмування змісту тексту;
- генерація відповідей на питання;
- логічний висновок (вилучення з тексту наявного сенсу і знань);
- переклад речень у аудіо (автоматичне озвучення тексту перекладу);
- переклад речень у відео

Далі у переліку можна виділити систему АОТ, яка дозволяє за допомогою браузеру проводити різні методи лінгвістичного аналізу.

Серед функціоналу системи:

- модуль графематичного аналізу тексту;
- компоненти морфологічного аналізу для рос., нім. і англ.яз .;
- модуль автоматичного знищення омонімії;
- модуль семантичного аналізу тексту;
- система лінгвістичного пошуку (конкорданс);
- різні тезауруси і словники.

Окрім того, що є можливість користуватися функціоналом системи у браузері є можливість також завантаження системи безкоштовно для Linux і Windows. Тексти програм для Linux доступні на умовах ліцензії LGPL.

Серед переваг системи можна виділити те, що є реалізація функціоналу в браузері та система є безкоштовною. Серед недоліків знову ж відсутність роботи з українською мовою.

Наступною системою, яка виконує аналіз тексту за допомогою статистичної обробки текстів є програма Ventli. На основі текстових елементів вона дає можливість підраховувати кількість параграфів, фраз, слів і знаків у файлі. На основі текстової статистики Ventile проводить вимірювання абсолютної частоти, три виміри середніх значень (моду, медіану і середнє арифметичне) і п'ять вимірів розкиду (мінімум, максимум, різниця квартилей, середнє відхилення, стандартне відхилення). Статистичні результати відтворюються у вигляді числової таблиці або графічно, у вигляді стовпчастої діаграми (гістограми).

Під час виконання аналізу було виявлено, що більша кількість систем, що реалізовані підтримує англійську мову. Було знайдено також системи, що підтримують російську мову. При цьому спостерігається значний дефіцит систем, що працюють з україномовними текстами. Результати порівняння систем можна переглянути у таблиці 1.

Для розробки системи аналізу тексту є можливість використовувати існуючі бібліотеки, що містять функціонал аналізу тексту. Однією з найбільш відомих бібліотек для аналізу тексту є NLTK. Це бібліотека на Python, що містить в собі велику кількість основних методів аналізу тексту та його попередньої обробки.

Для виконання аналізу тексту бібліотека використовує понад 50 лінгвістичних ресурсів, серед яких один з найбільш відомих WordNet.

Серед функціоналу бібліотеки можна виділити:

- графематичний аналіз (розбиття тексту на токени)
- морфологічний аналіз (визначення частина мови слова)
- синтаксичний аналіз (побудова дерева залежностей між словами відповідно до правил мови)
- семантичний аналіз (можливість визначення назв серед слів, серед яких абревіатури, назви компаній та установ і т.д)
- позначення слів (дозволяє додавати мета-інформацію до тексту)
- стемінг (процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс)
- лематизація (дозволяє зводити слова до спільнокореневих, що покращує аналіз тексту)

Реалізована бібліотека за допомогою мови Python. При цьому архітектура бібліотеки реалізована добре, так як вона використовується у багатьох додатках для розпізнавання тексту та постійно розширюється за допомогою open source розробників.

До переваг бібліотеки можна віднести те, що вона є безкоштовною та містить зручне API на python. Серед недоліків можно виділити те, що не реалізовано зручного користувацького інтерфейсу та є можливість працювати тільки з англійським текстом.

Наступною бібліотекою, що може використовуватися для аналізу тесту є spaCy. Серед функціоналу бібліотеки можна виділити:

- токенізація слів
- розпізнавання названої сутності
- підтримка великої кількості мов
- досліджені вектори слова

- тегування слів
- парсинг іменувань
- синтаксична сегментація речень
- вбудовані візуалізатори для синтаксису
- зручне хешування тексту
- експорт до масивів numpy

Gensim - бібліотека Python для визначення теми тексту, індексації документів та пошуку подібності між документами

Бібліотека дозволяє визначати тему тексту за допомогою ефективної реалізації наступних алгоритмів:

- Latent Semantic Analysis (LSA/LSI/SVD)
- Latent Dirichlet Allocation (LDA)
- Random Projections (RP)
- Hierarchical Dirichlet Process (HDP)
- word2vec deep learning.

Під час виконання аналізу було виявлено, що більша кількість систем, що реалізовані підтримує тільки англійську мову. При цьому була знайдена незначна частина систем російською мовою, які виконують окремий вид аналізу.

Під час виконання аналізу було виявлено, що більша кількість систем, що реалізовані підтримує тільки англійську мову. При цьому була знайдена незначна частина функціоналу російською мовою, який виконує окремий вид аналізу. При цьому цей функціонал не є повноцінною системою, а лише реалізацією певної функції.

Серед таких можна виділити російський веб сайт, що дозволяє виконувати морфологічний аналіз слова http://starling.rinet.ru/morph.htm. Після введення слова він дозволяє переглянути усі морфологічні характеристики слова залежно від частини мови. Наприклад, для дієслова він дозволяє переглянути інфінітив, усі його форми у однині й множині та у різних родах. Також він дозволяє переглянути форми слова у різних часах та у всіх відмінках.

Серед недоліків можна виділити реалізацію тільки однієї функції, а також підтримку тільки російського тексту.

Наступним серед окремих реалізацій функціоналу можна виділити додаток http://slashzone.ru/parser/, що виконує синтаксичний аналіз речення. Після введення речення та початку аналізу, парсер розбиває речення на слова та розмічує їх спеціальними тегами відповідного того, до якої частини мови вони відносяться. Після цього виводиться дерево залежностей між словами.

Серед недоліків знову можна виділити реалізацію тільки одного виду аналізу тексту, а також підтримку тільки російської мови.

1.4 Висновки до розділу

У даному розділі було виконано аналіз предметної області, проаналізовано існуючі системи для дослідження тексту, їх функціональність та проведено порівняльний аналіз.

Також було проаналізовано основні методи дослідження тексту та виділено серед них 3 основні класи: лінгвістичні методи, статистичні методи та методи з використанням машинного навчання.

Також було проаналізовано як відбувається аналіз тексту та основні етапи підготовки тексту до аналізу та його попередньої обробки.

На основі проаналізованих систем та методів було прийнято рішення які методи аналізу тексту потрібно реалізувати у системі, що розробляється.

2 ВИДИ МЕТОДІВ ДОСЛІДЖЕННЯ ТЕКСТУ

2.1 Аналіз тональності тексту

У роботі було реалізовано аналіз тональності українського тексту. У якості датасету використовувалися коментарі з фейсбук. Для збирання інформації використовувався парсер написаний на Javascript. Усі тексти, які збирав парсер фільтрувалися та обиралися тільки українські коментарі. Після цього було використано розмітку текстів за категоріями. Серед обраних категорій було два варіанти:

- позитивна
- негативна

Усього було зібрано 4267 коментарів для навчання моделі. Також існує можливість розширювати цей датасет, так як у системі реалізовано додавання тексту з можливістю його розмітки.

Далі після підготовки датасету було реалізовано його попередню обробку. Було реалізовано токенізацію тексту на речення та на слова. Після цього слова, що було отримано після токенізації було скорочено до їх початкової форми за допомогою обрізання кінцевих частин слів. Зведення до початкової форми дозволило покращити продуктивність аналізу українського тексту

Після попередньої обробки текст потрібно було привести до векторів чисел, так я алгоритми машинного навчання не можуть безпосередньо працювати з неочищеним текстом. Це називається виділенням ознак.

У роботі було використано модель “Bag of words”, яка я є популярною і проста технікою вилучення функцій, яка використовується під час роботи з текстом. Вона описує кожне слово у документі.

Для використання цієї моделі нам було створено словник відомих слів (їх також називають лексемами) та обрано міру присутності відомих слів

При цьому будь-яка інформація про порядок або структуру слів відкидался, саме тому ця модель називається “Bag of words”. Вона намагається зрозуміти чи трапляється в документі слово, але не знає де знаходиться це слово в документі.

Складність моделі полягала у вирішенні питання про те, як скласти словниковий запас відомих слів та як порахувати наявність відомих слів.

Коли розмір словникового запасу збільшувався, векторне представлення документів також збільшувалося. Таким чином довжина вектора документа дорівнювала кількості відомих слів.

У деяких випадках ми можемо мати величезну кількість даних, і в цьому випадку довжина вектора, який представляє документ, може становити тисячі або мільйони елементів. Крім того, кожен документ може містити лише кілька відомих слів у словниковому запасі.

Тому вектори матимуть багато нулів. Таким векторам з великою кількістю нулів потрібно більше пам’яті та обчислювальних ресурсів.

Ми можемо зменшити кількість відомих слів, коли використовуємо модель для зменшення необхідної пам’яті та обчислювальних ресурсів. Також потрібно використовувати методи очищення тексту, що були розглянуті раніше:

- ігнорування регістру
- ігнорування пунктуації
- видалення зайвих слів
- зведення слів до їх основної форми
- виправлення неправильно написаних слів

Також існує ще один більш складний спосіб створення словника - використання згрупованих слів. Це змінює обсяг словникового запасу і

дозволяє моделі отримати більш детальну інформацію про документ. Такий підхід називається n-грамами.

N-грами зазвичай відносяться до послідовності слів. Уніграма - це одне слово, біграма - це послідовність із двох слів, 3-грама - це послідовність із трьох слів і т.д. Позначення “n” у “n-грамі” означає кількість згрупованих слів.

Використання алгоритму із біграмами є більш ефективним

Після того, як створено словниковий запас наявних слів набуло оцінено кількість слів у даних.

Серед основних методів оцінки виділяють:

- бінарний підхід (1 для присутності, 0 для відсутності).
- підрахунок скільки разів кожне слово з’являється в документі.
- обчислення частоти появи кожного слова в документі у відношенні до всіх слів у документі.

У роботі було використано обчислення частоти появи кожного слова.

Однією із проблем, з якою ми зіштовхнулися під час підрахунку частоти слів стало те, що слова які найчастіше зустрічаються в документі починали мати найвищі оцінки. Ці слова не містили стільки інформаційного значення для моделі порівняно з деякими рідкісними і доменними словами.

Нами було використано один із підходів для вирішення цієї проблеми за допомогою пониження значення слів, які часто зустрічаються у всіх документах. Такий підхід називається TF-IDF. По суті це статистичне вимірювання, яке використовується для оцінки важливості слова для документа.

Значення підрахунку слова у TF-IDF збільшується пропорційно тому наскільки часто слово зустрічається у документі, але воно компенсується кількістю документів у групі, що містять слово.

Давайте розглянемо формулу, яка використовувалася для обчислення TF-IDF.

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

При цьому спочатку ми обчислювали наскільки слово часто зустрічається у даному документі

$$\mathrm{tf}(t, d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

Після чого обчислювали наскільки рідко слово зустрічається у інших документах.

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Після чого ми перемножували отримані значення.

Після векторизації тексту отримані значення було використано для навчання моделі. Навчання моделі було виконано за допомогою наївного баєсовського класифікатора.

Цей класифікатор є одним з найпопулярніших для аналізу тексту. Необхідною умовою роботи алгоритму є наявність набору даних для кожної категорії, яка використовується під час класифікації тексту

Цей метод класифікації, заснований на теоремі Байєса. Нами було виділено два класи: позитивний та негативний. Для того щоб класифікувати текст ми реалізували у алгоритмі підрахунок кількості разів, коли кожне з слів з’являлося в документі.

Ця методика добре працює для класифікації текстів, коли наявні різноманітні категорії.

Для документа d та класу c формула Байєса, яку було реалізовано виглядає наступним чином:

$$P(c | d) = [P(d | c) \times P(c)] / [P(d)]$$

Відображення класу для даного документа - це клас, який має максимальне значення вищезгаданої ймовірності.

Оскільки всі ймовірності мають P (d) як їх знаменник, ми можемо усунути знаменник і просто порівняти різні значення чисельника

$$P(c | d) = P(d | c) \times P(c)$$

Тепер представимо документ як сукупність слів x1, x2, x3,…

Тоді ми можемо переписати P (d | c) як:

$$P (x1, x2, x3,…, xn | c)$$

При цьому P (c) - загальна ймовірність класу, тобто як часто цей клас зустрічається загалом. Наприклад у нашому випадку позитивних і негативних, класів ми обчислювали ймовірність того, що будь-який даний огляд є позитивним або негативним, без фактичного аналізу поточного вхідного документа.

Це обчислюється шляхом підрахунку відносних частот кожного класу.

Наприклад з 10 відгуків, які ми бачили, 3 були віднесені до позитивних.

P (позитивний) = 3/10

Ми використовуємо деякі припущення для спрощення обчислення такої ймовірності:

- припущення, що положення слів у документі не має значення.

- умовна незалежність, яка припускає, що ймовірність слів P (xi | cj) не залежить одна від одної.

Важливо зазначити, що обидва ці припущення насправді не є коректними - зазвичай, порядок слів має значення, і вони не є незалежними.

Однак ці припущення значно спрощують складність обчислення ймовірності класифікації. І на практиці нами було обчислено ймовірность з розумним рівнем точності, враховуючи ці припущення.

Тобто для обчислення ймовірності P (d | c) x P (c) було обчислено P (xi | c) для кожного xi в d та помножено їх разом.

Потім помножено результат на P(c) для поточного класу. Також це потрібно зробити для кожного класу і обрати клас, який має максимальне загальне значення.

Щоб обчислити значення P(ci) ми використовували оцінки максимальної ймовірності, тобто ми дивилися на підрахунок частоти.

P(ci) = [Кількість документів, які були класифіковані як ci] / [Число документів]

Після обчислення було виявлено, що можуть виникати проблеми ящо слово не зустрічається жодного разу у документах. У цьому випадку P = 0

Оскільки ми обчислювали загальну ймовірність класу шляхом множення індивідуальних ймовірностей для кожного слова, ми отримали загальну ймовірність 0 для потрібного класу.

Для усунення цієї проблеми нами було використано алгоритм згладжування Лапласа.

Для цього ми змінили нашу умовну ймовірність слова, додаючи 1 до чисельника і змінюючи знаменник як такий:

$$P(wi \mid cj) = [count(wi, cj) + 1] / [\Sigma w \in V (count(w, cj) + 1)]$$

Це можна спростити до

$$P(wi \mid cj) = [count(wi, cj) + 1] / [\Sigma w \in V (count(w, cj)) + | V |]$$

де | V | - це наш розмір словникового запасу (ми можемо це зробити, оскільки додаємо по 1 для кожного слова в словник у попередньому рівнянні).

Таким чином нами було натреновано модель для аналізу тональності українського тексту. Після реалізації нашу модель було протестовано та провалідовано.

Для валідації моделі було виконано розбиття датасету на датасет для тренування та датасет для тестування. Розбиття виконувалися із оптимальним співвідношенням:

- датасет для тренування - 70%
- датасет для тестування - 30%

Після цього було реалізовано обчислення точності розпізнавання моделі на основі розмічених категорій. Після тренування моделі було отримано точність розпізнавання 80.7304% на датасеті для тестування, що є задовільним результатом. При цьому у системі була реалізована можливість довчати модель на основі нових даних за допомогою додавання нових розмічених текстів до датасету.

2.2 Частотний аналіз тексту

Під час виконання роботи нами було реалізовано частотний аналіз тексту, який дозволяє простежити стиль написання тексту та основі його закономірності.

Кількість різних літер в кожній мові обмежена і літери можуть бути просто перераховані, в тому числі і в українській мові. Важливими характеристиками тексту є повторюваність букв, біграм і взагалі n-грам сполучуваність букв один з одним, чергування голосних і приголосних і деякі інші. Ці характеристики є досить стійкими.

Реалізація полягала в підрахунку чисел входжень кожної можливої m-грами в досить довгих відкритих текстах T = t1t2 ... tl, складених з букв алфавіту {a1, a2, ..., an}. При цьому проглядалися m-грами тексту, із літер що йдуть підряд одна за одною:

$t_1t_2...t_m$, $t_2t_3...t_{m+1}$, ..., $t_{i-m+1}t_{l-m+2}...t_l$.

Якщо $\vartheta(a_{i1}a_{i2}...a_{im})$ - число появ m-грами ai1ai2 ... aim в тексті T, а L - загальне число підрахованих m-грам, то досвід показує, що при досить великих L частоти

$$\frac{\vartheta(a_{i1}a_{i2}...a_{im})}{L}$$

для даної m-грами мало відрізняються один від одного.

В силу цього, відносну частоту вважають наближенням ймовірності P (ai1, ai2 ... aim) появи даної m-грами у випадково обраному місці тексту (такий підхід прийнятий при статистичному визначенні ймовірності).

Приклад того як виглядає реалізований частотний аналіз у системі можна побачити на рисунку 2.1

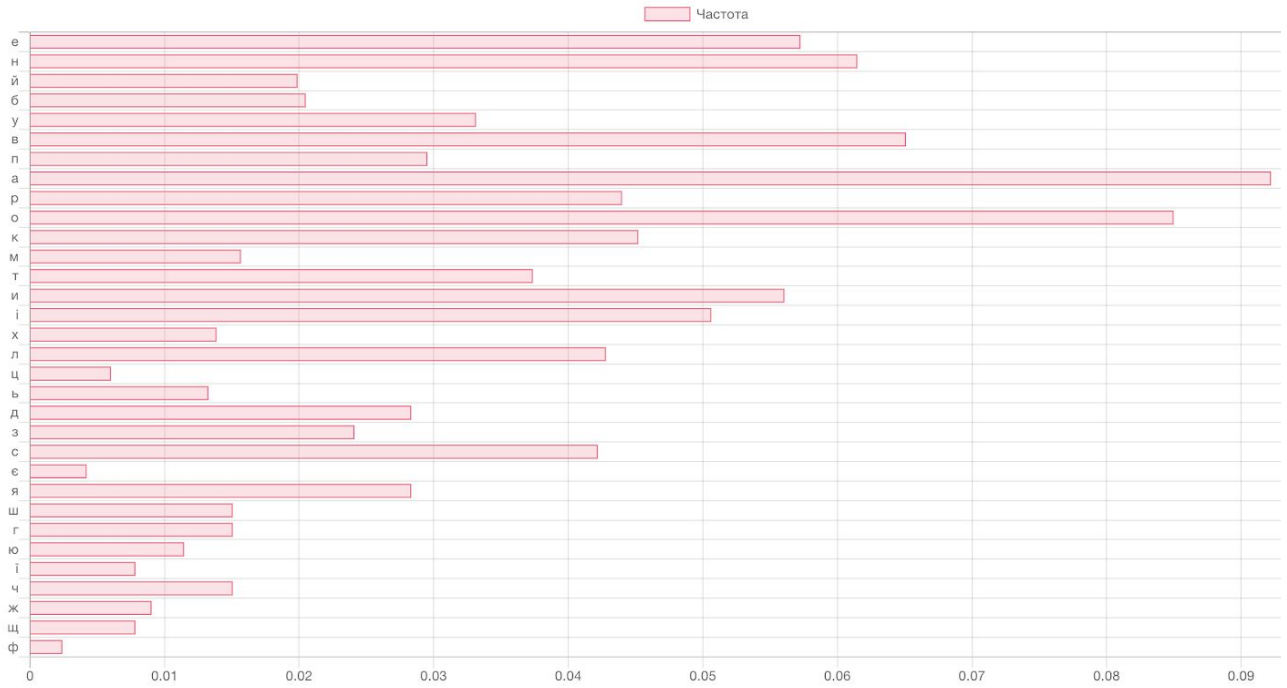


Рисунок 2.1 - Приклад частотного аналізу тесту, реалізованого у системі

Деяка різниця значень частот у наведених в різних джерелах таблицях пояснюється тим, що частоти істотно залежать не тільки від довжини тексту, а й від його характеру. Наприклад, в технічних текстах рідкісна буква Ф може стати досить частою в зв'язку з частим використанням таких слів, як функція, диференціал, дифузія, коефіцієнт і т.п.

Ще більші відхилення від норми в частоті вживання окремих літер спостерігаються в деяких художніх творах, особливо у віршах. Тому для надійного визначення середньої частоти букв бажано мати набір різних текстів, запозичених з різних джерел. Разом з тим, як правило, подібні відхилення незначні, і в першому наближенні ними можна знехтувати.

Для отримання більш точних відомостей про відкриті текстах можна будувати і аналізувати таблиці біграми.

Корисною є інформація про сполучуваність букв, тобто про бажаних зв'язках букв один з одним, яку легко витягнути з таблиць частот біграм. Мається на увазі таблиця, в якій зліва і праворуч від кожної букви розташовані найкращі "сусіди" (в порядку убування частоти відповідних биграмм). У таких

таблицях зазвичай вказується також частка голосних і приголосних букв (у відсотках), що передують даній букві або знаходяться після неї. Приклад реалізованого частотного аналізу біграм можна спостерігати на рисунку 2.2.

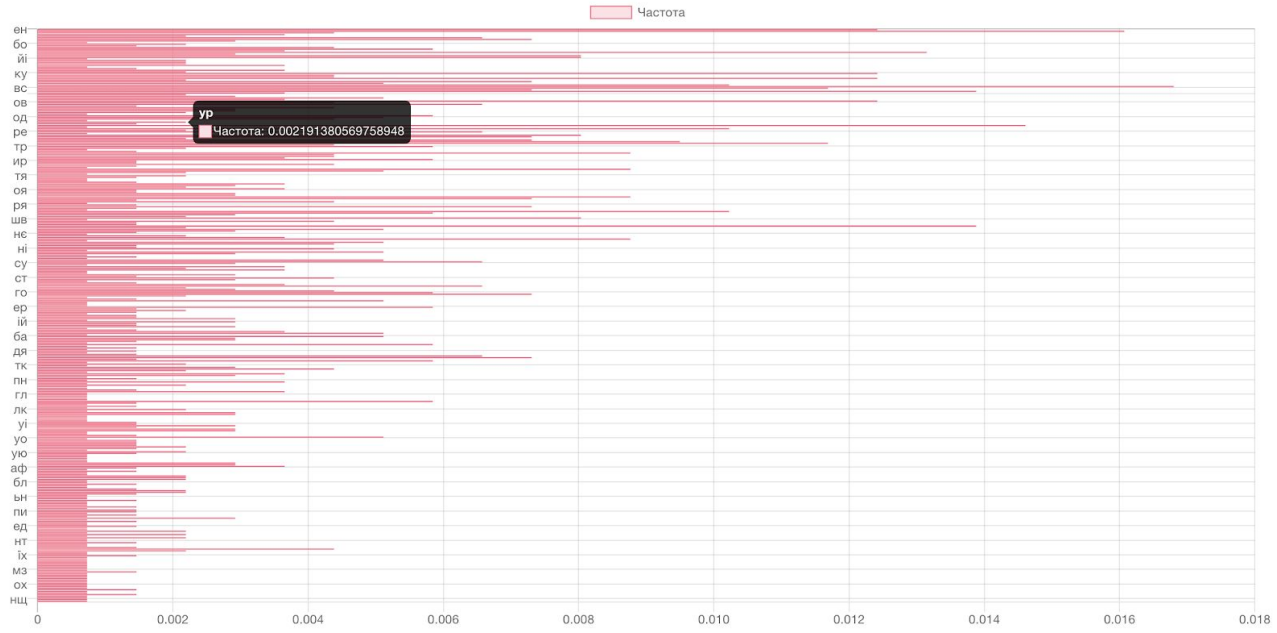


Рисунок 2.2 - Частотний аналіз тексту за допомогою біграм

При аналізі сполучуваності букв між собою слід мати на увазі залежність появи літер у відкритому тексті від значного числа попередніх літер. Для аналізу цих закономірностей використовують поняття умовної ймовірності.

Визначення закономірностей за допомогою статистичного аналізу відіграє велику роль у криптоаналізі. Зокрема, використовується при побудові формалізованих критеріїв на відкритий текст, що дозволяють застосовувати методи математичної статистики в задачі розпізнавання відкритого тексту в потоці повідомлень. При використанні ж спеціальних алфавітів потрібні аналогічні дослідження частотних характеристик текстів, що виникають, наприклад, при машинному обміні інформацією або в системах передачі даних. У цих випадках побудова формалізованих критеріїв для тексту завдання значно складніше.

Крім криптографії частотні характеристики відкритих повідомлень істотно використовуються і в інших сферах. Наприклад, клавіатура комп'ютера,

друкарської машинки або лінотипу - це чудове втілення ідеї прискорення набору тексту, пов'язане з оптимізацією розташування букв алфавіту відносно один одного в залежності від частоти їх застосуванню.

2.3 Висновки до розділу

У даному розділі було реалізовано метод аналізу з використанням машинного навчання. Для реалізації було обрано розпізнавання тональності тексту за допомогою наївного байєсівського класифікатора. Датасет було підготовлено з використанням парсеру для facebook, після чого розмічено. Точність розпізнавання натренованої моделі становить 80.7304%, що є достатньо гарним результатом.

Окрім цього у даному розділі було також реалізовано статистичний аналіз тексту. Для статистичного аналізу тексту було обрано реалізувати частотний аналіз, який було виконано для літер, а також для біграм тексту. Результати було відображено за допомогою візуалізації.

3 ОРГАНІЗАЦІЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ТЕКСТУ

3.1 Організація модулів системи

Основною метою проектування системи є її масштабування. Тобто система була спроектована таким чином, що в неї можна з легкістю можна додавати різні модулі для аналізу тексту. Серед основних модулів було виділено:

- модуль з методами аналізу за допомогою машинного навчання
- модуль з методами аналізу за допомогою статистики
- модуль лінгвістичного аналізу

У свою чергу модуль лінгвістичного аналізу включає морфологічний аналіз

На рисунку 3.1 можна переглянути основні сценарії використання системи

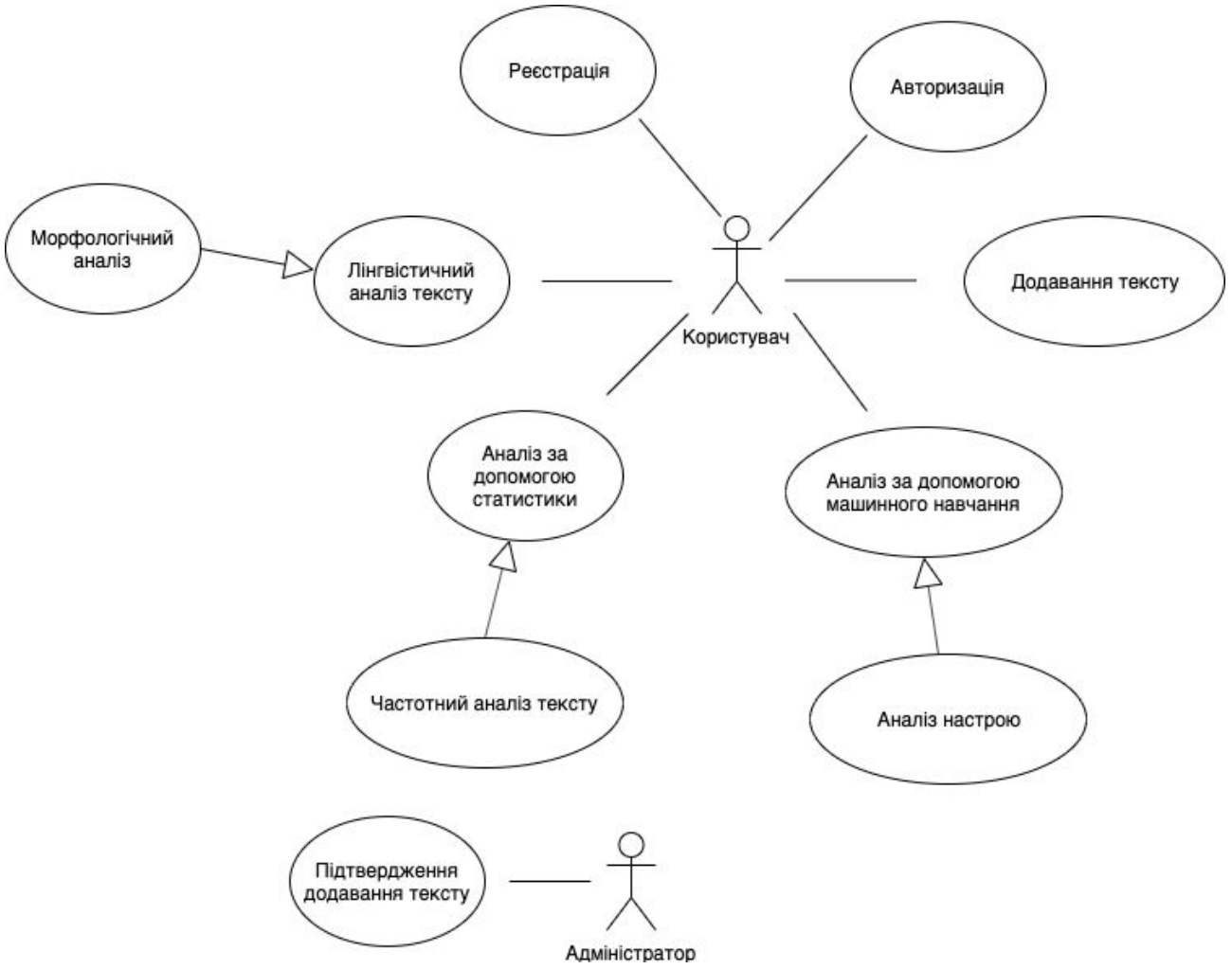


Рисунок 3.1 - Діаграма сценаріїв використання системи

Для того, щоб працювати з системою користувач повинен авторизуватися. Якщо користувач ще не зареєстрований у системі, то він має змогу зареєструватися та почати свою роботи з системою. Для того, щоб зареєструватися користувач повинен ввести свій email, ім’я, фамілію та пароль. Після цього користувачу потрібно ввести свій email та пароль і його буде авторизовано у системі.

Авторизацію користувача було реалізовано за допомогою токенів. Під час авторизації користувач отримує токен, який використовується при кожному API запиті на сервер для того, щоб ідентифікувати користувача. При цьому так як використовувався JWT (JSON Web Token) у токені є можливість зберігати корисну інформацію. Наприклад, у системі було реалізовано зберігання id юзера у токені для отримання інформації про поточного користувача. Під час

кожного запиту на сервер виконується перевірка валідності токену, що прийшов від користувача. При цьому якщо токен прийшов невалідний, то сервер повертає запит із статусом 401 Unauthorized.

Для того, щоб почати аналіз тексту користувач повинен мати змогу його дати. Для цього реалізовано модуль додавання тексту, у якому користувач може написати свій текст через редактор або загрузити його за допомогою файлу текстового формату. Серед форматів, що підтримуються це файли .txt та .doc формату. Після того, як користувач додав текст він зберігається на сервері та має змогу використовувати його у своїх дослідженнях. При цьому є можливість зробити текст публічним або приватним. Публічні тексти після додавання є видимими для інших користувачів та можуть використовуватися ними у своїх дослідженнях. Приватні тексти є видимими тільки для користувача, який їх додав до системи.

3.2 Спроектовані сутності системи

У якості бази даних була обрана база даних MySQL. При цьому для роботи з базою даних була використана бібліотека, що допомагає представляти сутності об’єктами та має назву Sequelize і дозволяє зручно працювати з сутностями системи.

Сутності системи було спроектовано таким чином, щоб систему можна було максимально розширювати. Переглянути діаграму сутностей системи можна на рисунку 3.2.

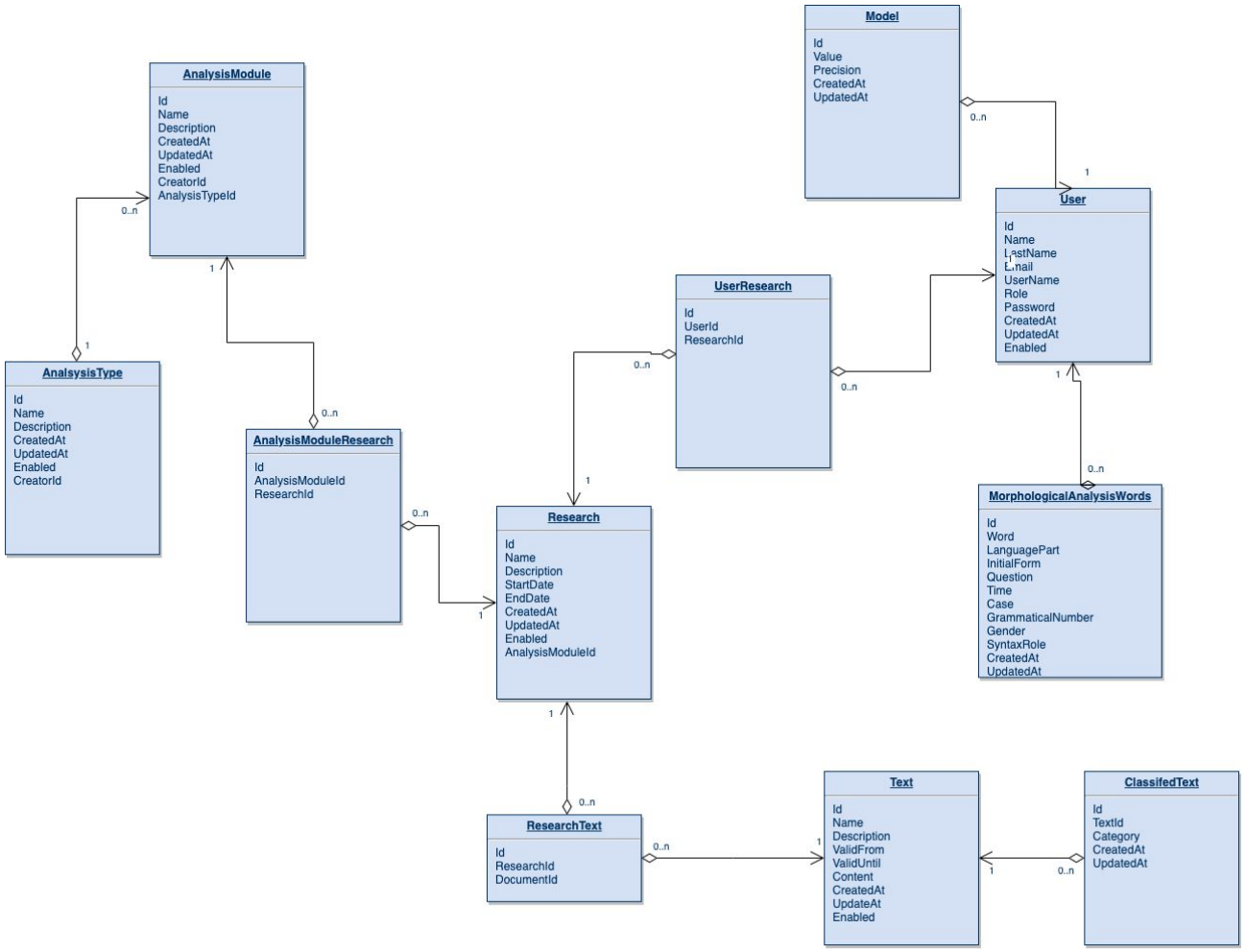


Рисунок 3.2 - Діаграма сутностей системи

Уся інформація про користувача зберігається у таблиці User. Після того як користувач зареєструвався уся введена ним інформація додається до цієї таблці. При цьому кожен користувач має змогу додавати дослідження, які зберігаються у таблиці Research. Кожне дослідження містить його назву, опис,

дату початку та дату закінчення. Після того як користувач створив дослідження він має змогу додавати інших користувачів до цього дослідження. Таким чином користувачі мають змогу виконувати колективні дослідження. Саме тому відношення між таблицями Research та User є many-to-many. Так як кожен користувач може мати декілька досліджень, у яких він бере участь. У свою чергу дослідження також можуть мати декілька користувачів, які приймають у ньому участь. Таблицею, що поєднує ці таблиці є таблиця UserResearch.

На діаграмі можна побачити, що користувач має змогу додавати модулі, які будуть зберігатися у таблиці AnalysisModule. При цьому кожен модуль належить до одного із типів аналізу, який зберігається у таблиці AnalysisType.

Користувач має змогу додавати модулі до свого дослідження. При цьому кожне дослідження може мати по декілька використаних модулів у цьому дослідженні. Модулі з аналізами можуть використовуватися у багатьох дослідження. Саме тому зв’язок між цими таблицям many-to-many. Таблицею, яка поєднує ці дві таблиці є AnalysisModuleResearch.

3.3 Основні архітектурні компоненти системи

Для того, щоб дослідники лінгвісти могли з легкістю використовувати систему вона була реалізована як веб додаток. Таким чином дослідникам лінгвістам не потрібно встановлювати її на своєму комп’ютері, а все що від них вимагається - це наявність веб браузера.

При розробці додатку було використано розділення клієнтської та серверної частини на окремі частини, які взаємодіють між собою за допомогою REST API.

Для розробки веб-додатку була обрана мова програмування JavaScript як для клієнтської частини, так і для серверної, що дозволяє писати уніфікований код та робить контракти між сервером та клієнтом більш зручними

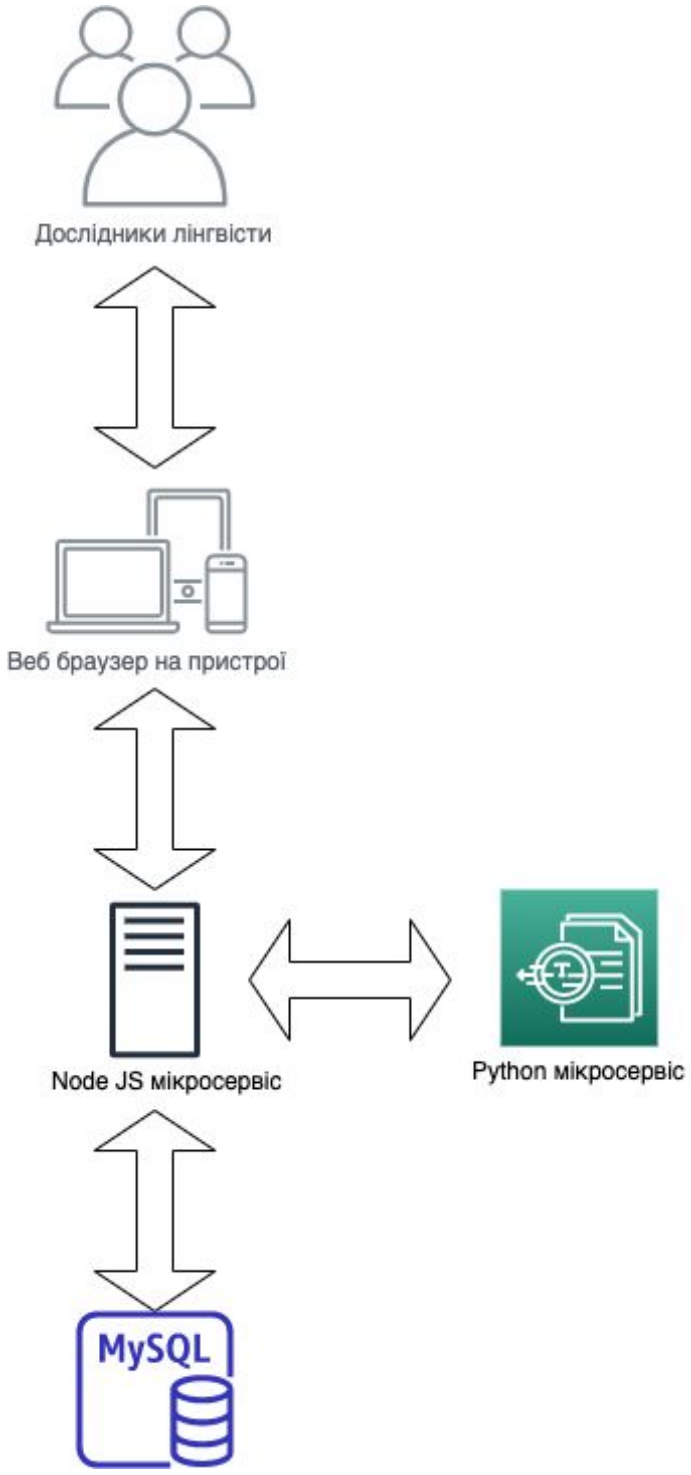


Рисунок 3.3 - Основні складові системи

Для клієнтської частини системи було взято підхід Single Page Application. Цей підхід використовується у багатьох додатках та дозволяє розділити серверну та клієнтську частини на окремі складові й дозволяє окремо вести розробку клієнтської та серверної частин.

Коли користувач переходить з однієї сторінки на іншу в односторінкових додатках, то вся зручність полягає у тому, що сторніка кожен раз не перезавантажується, підміняються лише її окремі складові за рахунок нових даних. Це дуже схоже на те, як відбувається навігація у програмах, що встановлюються на комп'ютері.

Кожен з таких додатків повинен складатися з наступних частин:

- клієнтський роутинг
- шаблонізатор
- API для серверної частини
- завантаження даних за допомогою ajax

При цьому існує велика кількість бібліотек та фреймворків для розробки додатків з використанням SPA. Серед найбільш популярних можна виділити React, Angular, Vue JS.

Для розробки системи було обрано React, так як на даний момент він є найбільш популярним та має найбільш широку підтримку розробників, зокрема розробників Facebook, які використовують цю бібліотеку для розробки Facebook, тому є найбільш активними у підтримці розробки даної бібліотеки. Також ця бібліотека має найкращу продуктивність на даний момент серед бібліотек, що існують за рахунок реалізації алгоритму Virtual DOM. Virtual DOM – це алгоритм, який дозволяє не перемальовувати кожен раз окремі складові сторінки, якщо у цьому не має потреби. Це реалізовано за рахунок того, що всі складові компоненти користувацького інтерфейсу представлені у вигляді дерева об'єктів у JavaScript. З рахунок цього є можливість порівнювати чи змінився об'єкт у JavaScript, після чого приймати рішення про те чи потрібно його перемальовувати чи ні. Порівняння у JavaScript відбувається набагато скоріше, ніж операції у DOM, що й дозволяє додаткам на React працювати набагато швидше.

React відповідає лише за відображення користувацького інтерфейсу. Для роботи з даними у додатку була використана бібліотека Redux. Вона дуже часто використовується саме з додатками на React, але при цьому є популярною і для інших фреймворків. Бібліотека значно спрощує режим відлагодження та знаходження помилок, так як усі дані знаходяться в спільному дереві в одному джерелі.

У Redux взаємодія з даними відбувається на одному верхньому рівні. При цьому для того, щоб оновити дані потрібно зробити певну подію, у термінології Redux – dispatch. При цьому сама подія не змінює дані, вона лише надає повідомлення про те, що потрібно змінити. А оновлення здійснює чиста функція – reducer. Чиста функція – це функція, що отримує певні аргументи та повертає результат без оновлення параметрів та даних додатку. Чисті функції є одним з основних підходів у функціональному програмуванні.

Redux має три головних правила, які складаються з:

- єдиного джерела правди
- стана лише для читання
- оновлення даних проходить за допомогою чистих функцій

Також усі зміни у Redux відбуваються в одному напрямку і не можуть бути двосторонніми (Рисунок 3.4).

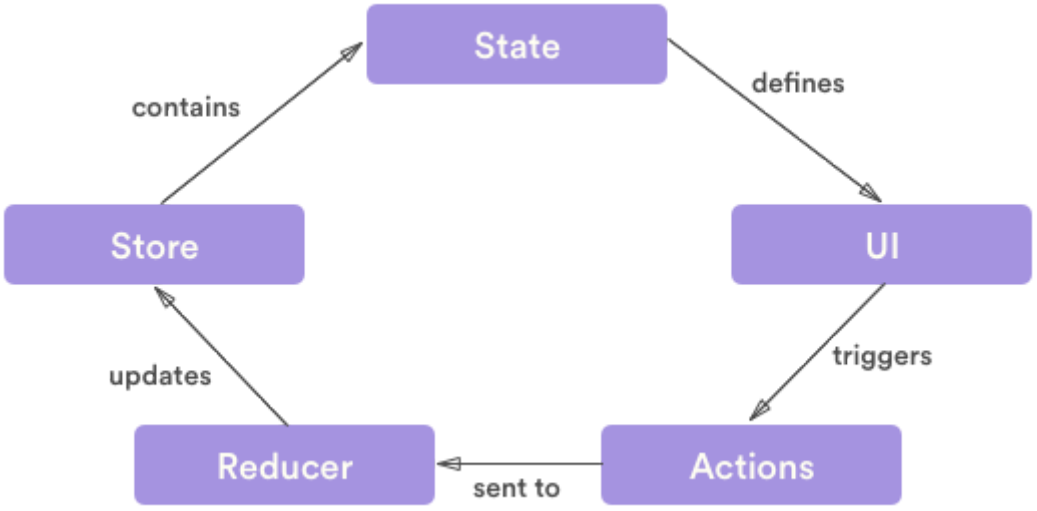


Рисунок 3.4 Архітектура Redux

На рисунку видно, що може відбутись якась подія, наприклад отримано дані з API. Далі про цю дію повідомляється у додатку за допомогою dispatch. Dispatch повідомляє стан додатку про подію, після чого дані оновлюються за допомогою чистих функцій, а зміна даних є причиною оновлення певних компонентів додатку. Тим часом компоненти додатку складатися подій, що породжують нові дії (actions). Тобто усі змінити виконуються у одному напрямку.

За допомогою цього архітектура додатку є більш прозорою та менш схильною до породження помилок через складність роботи зі станом додатку, так як виключені складні перехресні зв’язки між окремими частинами додатку, через що виникають непередбачувані помилки у системі.

Для розробки серверу додатку було обрано Node JS. Node JS – це програмна платформа, яка використовує V8 - рушій Javascript з відкритим сирцевим кодом, який було написано корпорацією Google. Він містить у собі величезну кількість оптимізацій, за рахунок чого платформа працює з достатньо високою продуктивністю. Також у Node JS є реалізація модулів, написаних на

C++ для взаємодії з операційною системою (операції з файлами та з мережею і т.д).

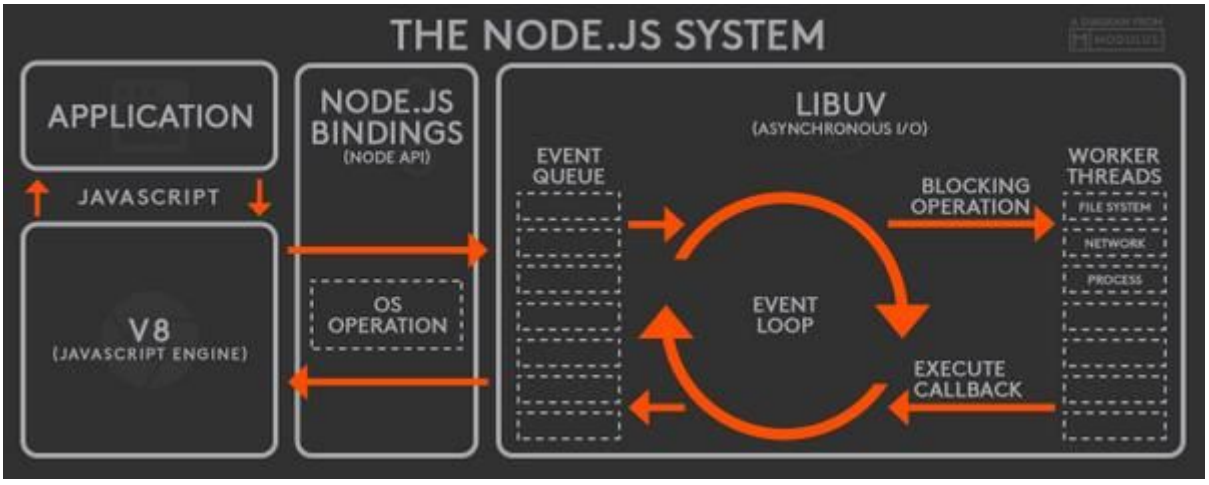


Рисунок 3.5 – Архітектура платформи Node JS

На рисунку 3.5 можна побачити основні компоненти, з яких складається Node JS. Платформа містить у собі концепції подійно-орієнтованого і асинхронного програмування з вводом/виводом, який є неблокуючим.

У подійно-орієнтованому програмуванні деякі частини системи можуть підписуватися на події та виконувати дії, коли ці події відбуваються, тобто реагувати на ці події. Коли певна подія трапляється, то всі підписники повідомляються, про те що ця подія відбулася за допомогою повідомлення, яке може містити певні дані. За допомогою такого підходу модулі взаємодіють між собою. По суті це є реалізацією шаблону проектування ”Спостерігач”.

Ввод/вивод у Node JS є неблокуючим, так як усі події є асинхронними та не блокують виконання основного потоку програми. Тобто коли відбувається операція вводу/виводу то основна програма не призупиняє своє виконання, а продовжує далі. Про те що операція вводу/виводу основна програма повідомляється за допомогою callback - функцій, які визиваються після того як операція вводу/виводу закінчилася. У Node JS вся організація асинхронного коду відбувається за допомогою бібліотеки libuv (рисунок 3.6)

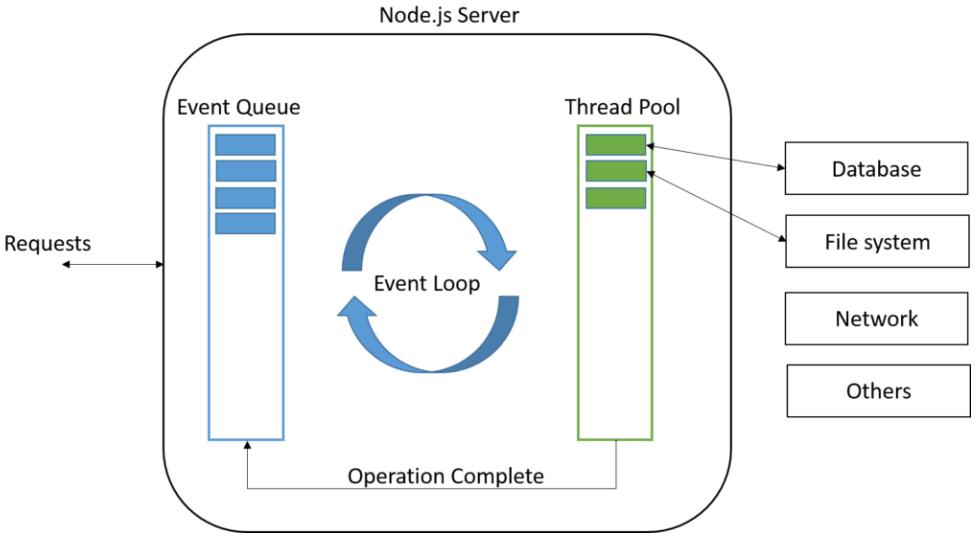


Рисунок 3.6 – Архітектура libuv

Саме завдяки такій архітектурі Node JS можна обробляти велику кількість запитів одночасно та показувати високу продуктивність та організовувати асинхронну роботу з кодом.

3.4 Висновки до розділу

У даному розділі було описано основні модулі, які було спроектовано у системі. Окрім цього було проаналізовано основні сценарії використанням системи та розроблено діаграму сценаріїв використання системи.

Також було спроектовано основні сутності системи та організовано взаємодію між ними таким чином, щоб систему можна було достатньо легко масштабувати.

Далі було описано основні архітектурні компоненти системи та взаємодію між ними. Також було обрано технології для реалізації спроектованих архітектурних компонентів та описано які переваги у спроектованій системі надають ці технології.

4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ АНАЛІЗУ ТЕКСТУ

4.1 Опис програмного забезпечення

Для аналізу тексту було розроблено веб-додаток для того, щоб кожен користувач міг зайти у додаток за допомогою браузера.

Клієнт розроблений за допомогою Javascript із використанням бібліотеки React JS. Серверну частину було розроблено на Node JS із використанням Next JS для server side rendering. Для API було використано фреймворк Express JS.

Також було розроблено окремий мікросервіс на Python, який за допомогою машинного навчання виконує аналіз тексту. Спілкування із мікросервісом Python відбувається за допомогою API.

У якості бази даних було обрано базу даних MySQL. Основні сутності системи можна переглянути у таблицях 4.1 - 4.5.

Таблиця 4.1 – Опис сутності User

Стовбець	**Тип**	**PK**	**Опис**
id	INT	+	Унікальний ідентифікатор
firstName	VARCHAR	-	Ім’я користувача
lastName	VARCHAR	-	Фамілія користувача
email	VARCHAR	-	Пошта
password	VARCHAR	-	Пароль
createdAt	DATETIME	-	Дата створення
updatedAt	DATETIME	-	Дата оновлення
enabled	Boolean	-	Доступність

Таблиця 4.2 – Опис сутності Research

Стовбець	**Тип**	**РК**	**Опис**
id	INT	+	Унікальний ідентифікатор
name	VARCHAR	-	Назва дослідження
description	TEXT	-	Опис
startDate	DATETIME	-	Дата початку
endDate	DATETIME	-	Дата закінчення
createdAt	DATETIME	-	Дата створення
updatedAt	DATETIME	-	Дата оновлення
enabled	Boolean	-	Доступність

Таблиця 4.3 – Опис сутності Document

Стовбець	**Тип**	**РК**	**Опис**
id	VARCHAR	+	Унікальний ідентифікатор
name	VARCHAR	-	Назва документу
description	TEXT	-	Опис
content	LONGTEXT	-	Зміст
createdAt	DATETIME	-	Дата створення
updatedAt	DATETIME	-	Дата оновлення
enabled	Boolean	-	Доступність

Таблиця 4.4 – Опис сутності AnalysisModule

Стовбець	**Тип**	**РК**	**Опис**
id	INT	+	Унікальний ідентифікатор
name	VARCHAR	-	Назва модулю
description	TEXT	-	Опис
createdAt	DATETIME	-	Дата створення
updatedAt	DATETIME	-	Дата оновлення
enabled	Boolean	-	Доступність
creatorId	INT	-	Id користувача, що створив
analysisTypeId	INT	-	Id типу аналізу

Таблиця 4.5 – Опис сутності AnalysisType

Стовбець	**Тип**	**РК**	**Опис**
id	INT	+	Унікальний ідентифікатор
name	VARCHAR	-	Назва типу аналізу
description	TEXT	-	Опис
createdAt	DATETIME	-	Дата створення
updatedAt	DATETIME	-	Дата оновлення
enabled	Boolean	-	Доступність

Продовження таблиці 4.5

creatorId	INT	-	Id користувача, що створив

4.2 Організація функціоналу додатку

Для виконання аналізу тексту дослідниками-лінгвістами було розроблено веб-додаток. Коли користувач вперше заходить у додаток, то він потрапляє на сторінку авторизації, яку можна побачити на рисунку 4.1

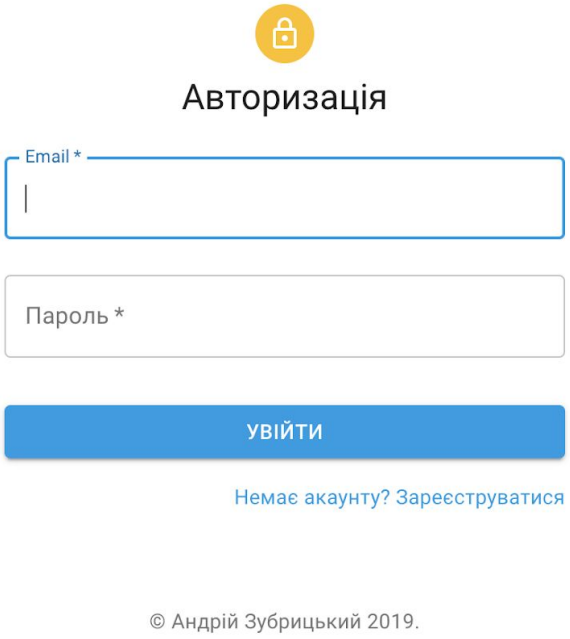


Рисунок 4.1 - Сторінка авторизації користувача

Для авторизації користувача він повинен ввести свій email та пароль, після чого його буде авторизовано у систему. Якщо користувач немає акаунту, то він має змогу реєстрації, натиснувши кнопку “Зареєструватися”. Після цього перед ним з’явиться сторінка реєстрації, яку можна переглянути на рисунку 4.2

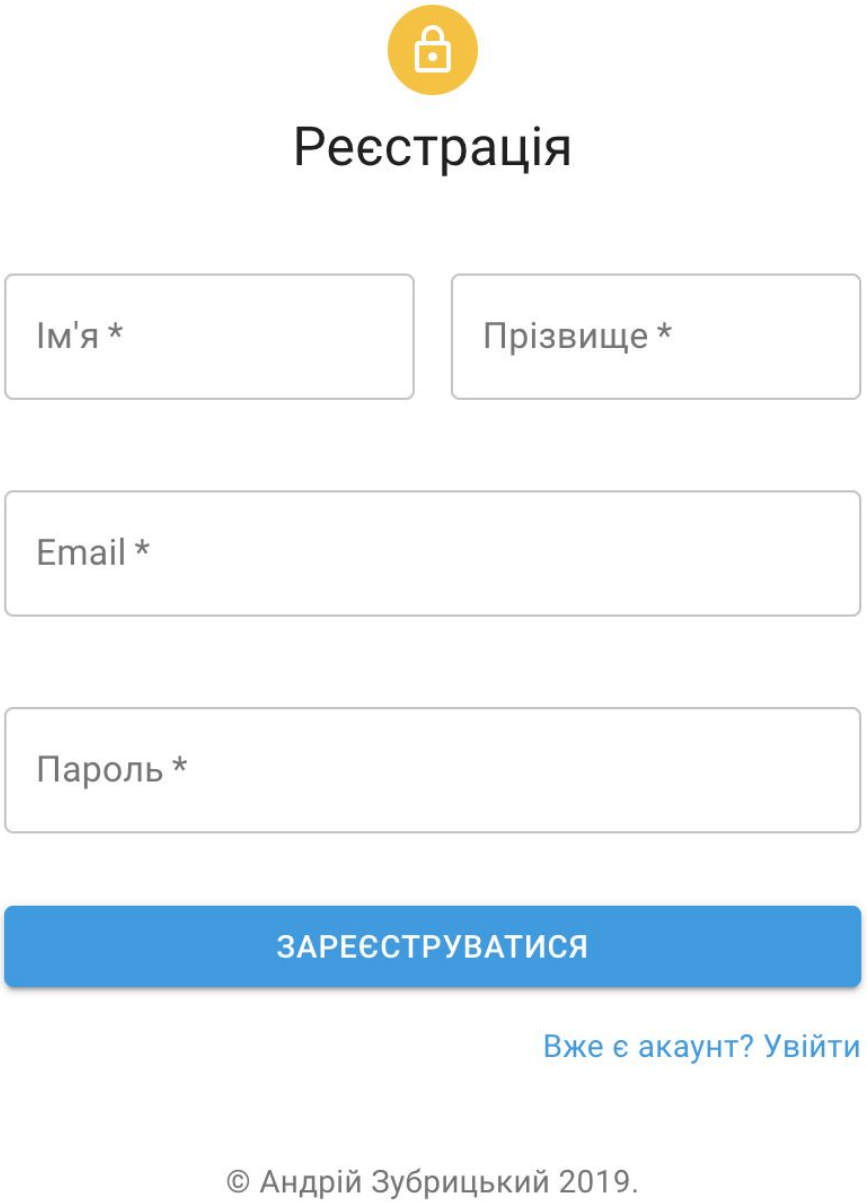


Рисунок 4.2 - Сторінка реєстрації користувача

Після того як користувач зареєструвався його буде перенаправлено на головну сторінку додатку, на якій за замовчуванням знаходиться перший метод лінгвістичного аналізу - морфологічний аналіз. На рисунку 4.3 можна побачити сторінку морфологічного аналізу.

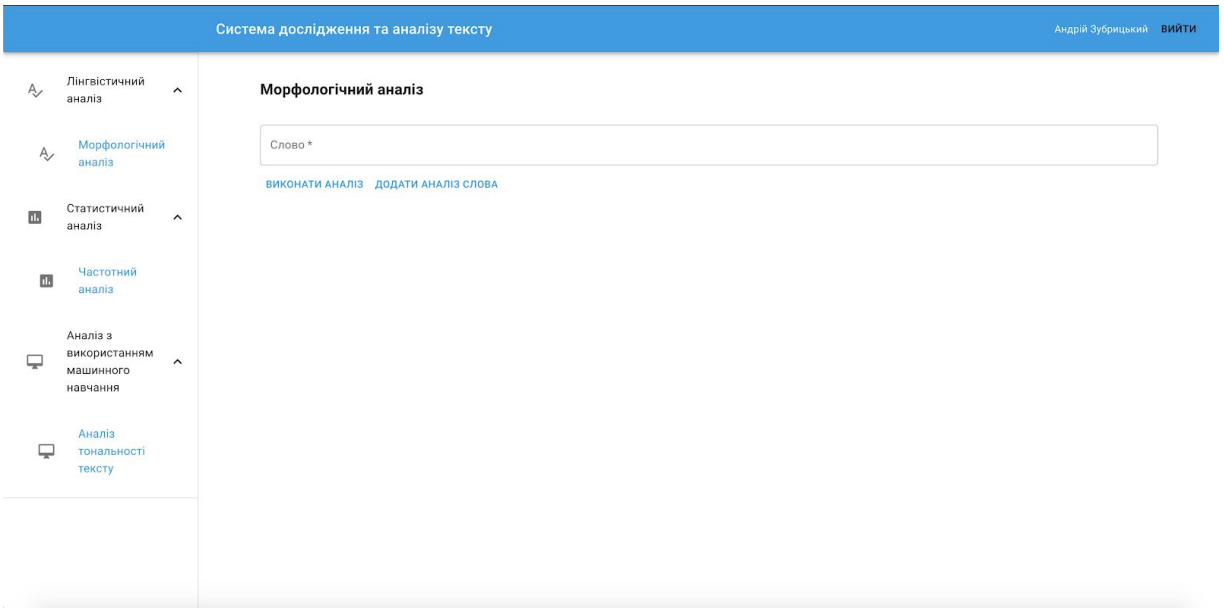


Рисунок 4.3 - Сторінка морфологічного аналізу

На сторінці морфологічного аналізу користувач має змогу ввести слово та отримати його морфологічний аналіз, натиснувши кнопку “Виконати аналіз”. Приклад можна переглянути на рисунку 4.4

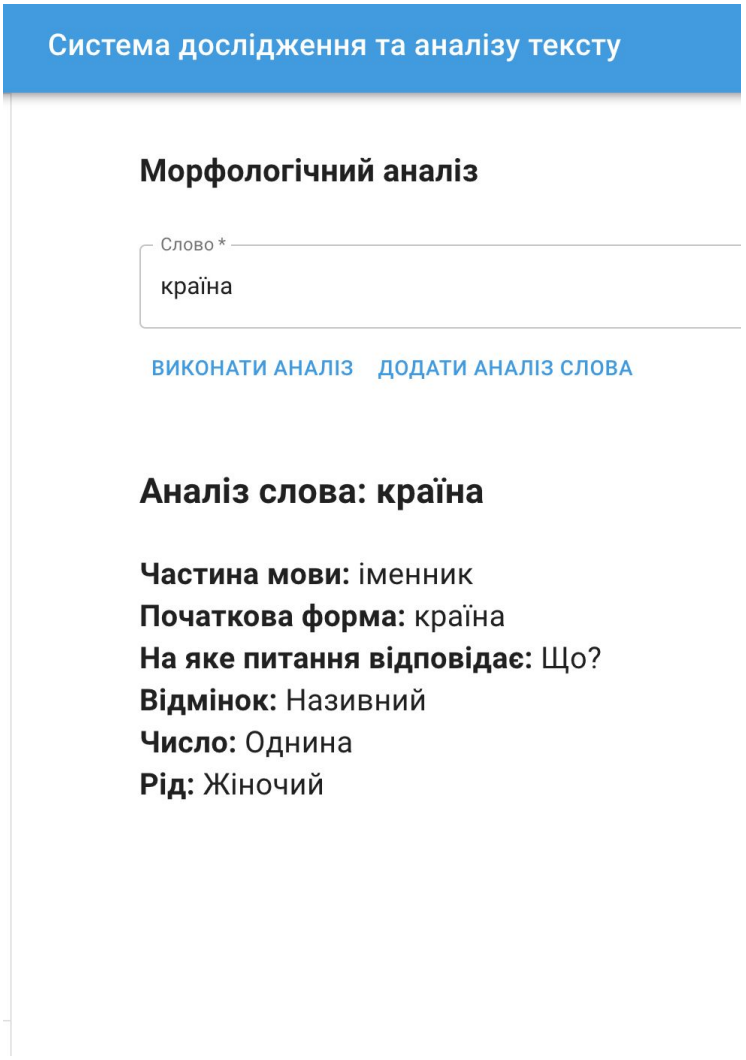


Рисунок 4.4 - Приклад морфологічного аналізу слова

Якщо морфологічного аналізу для заданого слова не було знайдено, то користувач має змогу додати аналіз слова натиснувши кнопку “Додати аналіз слова”, після чого його буде перенаправлено на сторінку додавання аналіз слова, яку можна переглянути на рисунку 4.5

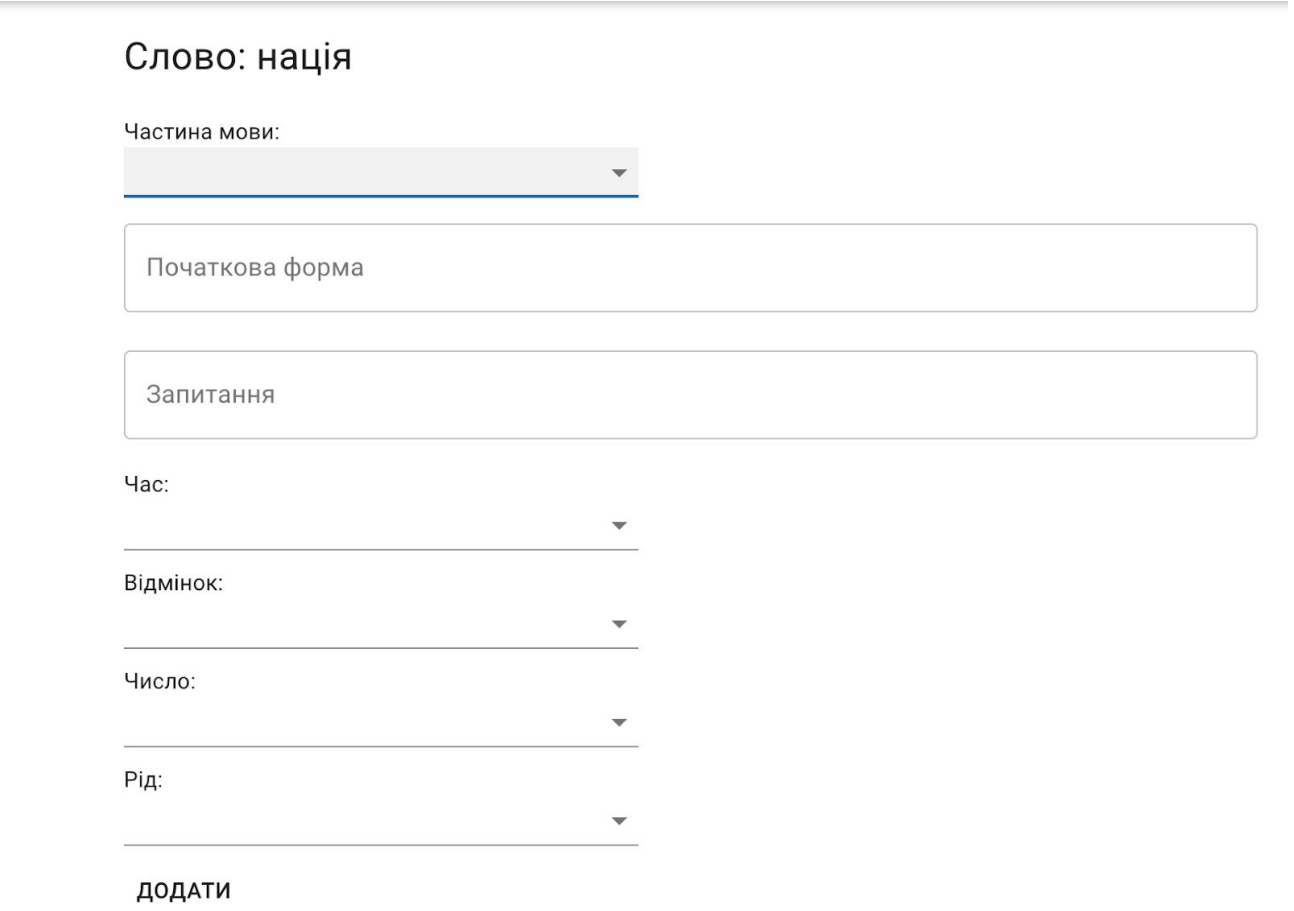


Рисунок 4.5 - Сторінка додавання морфологічного аналізу слова

Також користувач має змогу виконати частотний аналіз слова, для чого його потрібно обрати пункт меню “Статистичний аналіз” та підпункт “Частотний аналіз”. Після цього перед користувачем буде сторінка частотного аналізу, на якій потрібно ввести текст для виконання аналізу (рисунок 4.6).

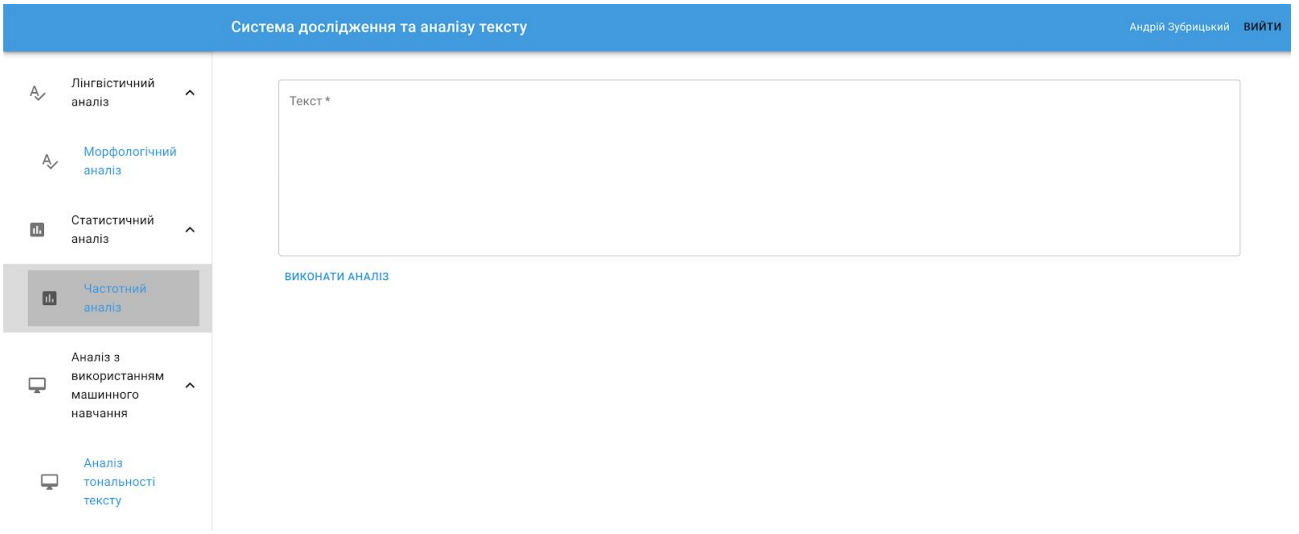


Рисунок 4.6 - Сторінка частотного аналізу тексту

Після того як текст додано потрібно нажати кнопку “Виконати аналіз”, після чого користувач отримує частотний аналіз літер тексту, який зображений у виді графіків (рисунок 4.7)

Рисунок 4.7 - Частотний аналіз літер тексту

Також користувач отримує частотний аналіз біграм тексту, приклад якого можна переглянути на рисунку 4.8.

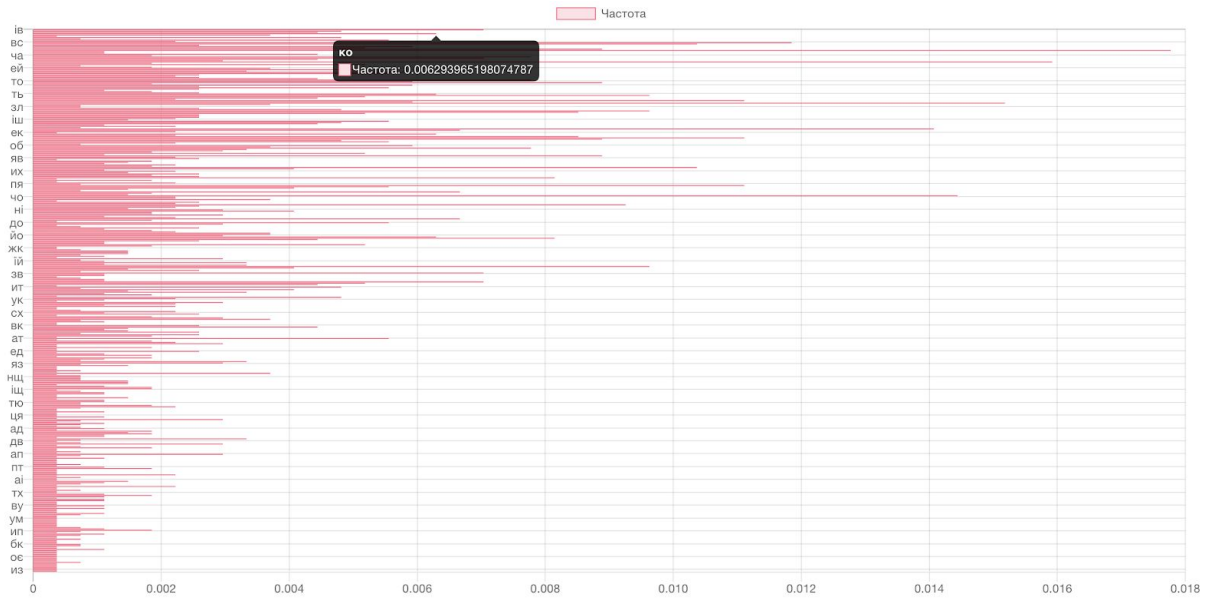


Рисунок 4.8 - Частотний аналіз біграм тексту

Окрім графіків користувач також отримує інформацію про кількість та частоту кожної літери та біграми у тексті. Ця інформація представлена у виді таблиці, приклад якої можна переглянути на рисунку 4.9.

Букви

Буква	Кількість	Частота
і	161	0.05
в	223	0.07
а	303	0.09
н	177	0.05
к	139	0.04
о	273	0.08
т	147	0.04
л	144	0.04
я	95	0.03

Рисунок 4.9 - Таблиця частотного аналізу

Також користувач має змогу виконати аналіз тональності тексту за допомогою машинного навчання. Для цього йому потрібно обрати пункт меню “Аналіз з використанням машинного навчання” та підпункт “Аналіз тональності тексту”. Після цього перед користувачем з’явиться сторінка аналізу тональності тексту (рисунок 4.10)

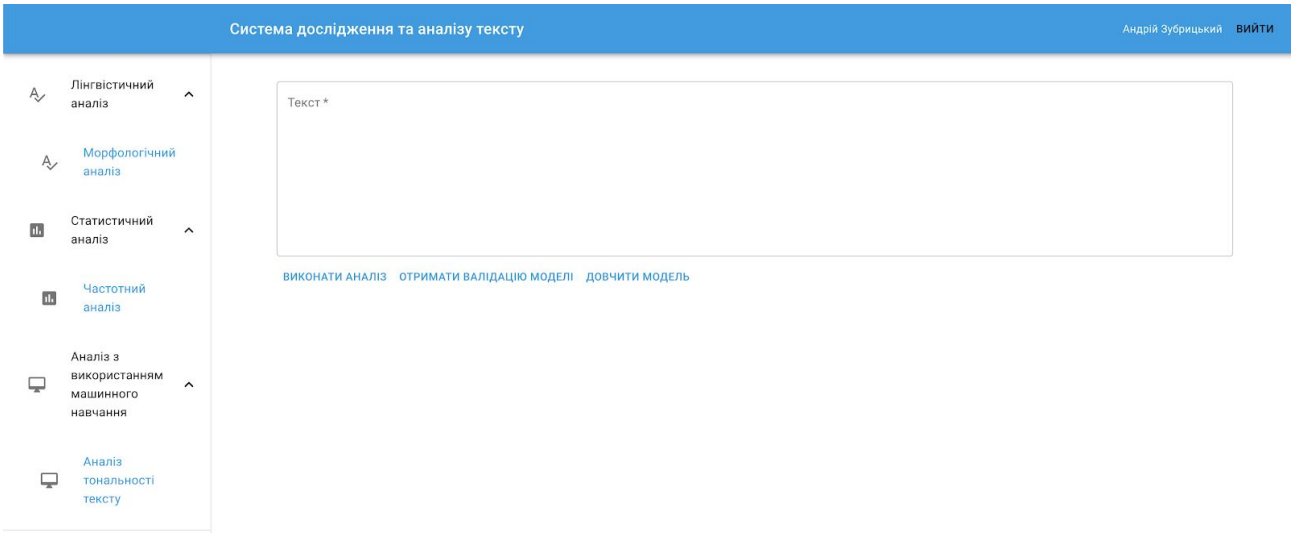


Рисунок 4.10 - Сторінка аналізу тональності тексту

Для того, щоб розпізнати тональність тексту користувачу потрібно ввести текст та натиснути кнопку “Виконати аналіз”. Після цього користувач отримує результати аналізу з ймовірністю, з якої натренована модель визначила категорію.

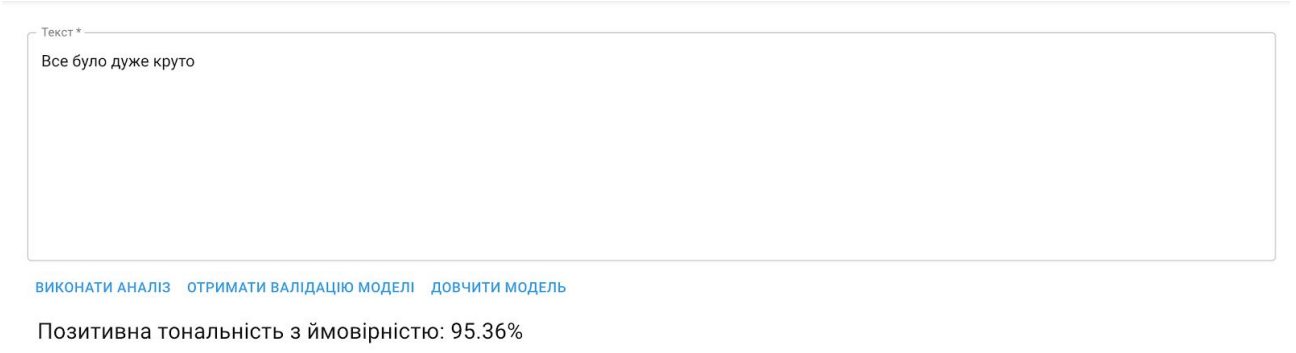


Рисунок 4.11 - Результати розпізнавання тональності тексту

Якщо результати розпізнавання не задовольнили користувача, то він має змогу довчити модель додавши дані до датасету, натиснувши кнопку “Довчити модель”. Після цього користувачу буде запропоновано ввести текст та обрати категорію (позитивна чи негативна), до якої належить текст (рисунок 4.12).

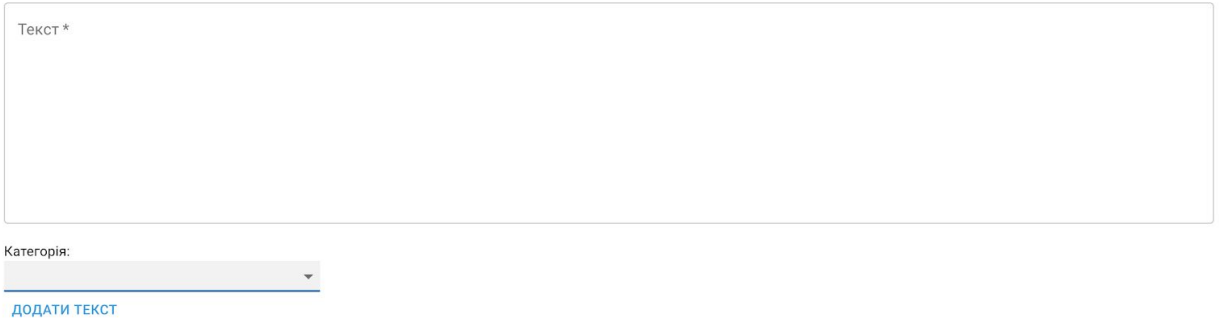


Рисунок 4.12 - Сторінка додавання тексту до датасету

Після того, як користувач додав текст модель буде натреновано на новому тексті, що дає у перспективі можливість покращити результати розпізнавання.

Також користувач має можливість отримати результати розпізнавання моделі на тестовому датасеті на даний момент. Для цього на сторінці аналізу тональності тексту потрібно натиснути кнопку “Отримати валідацію моделі”. Після цього користувач отримає результати валідації моделі (рисунок 4.13).

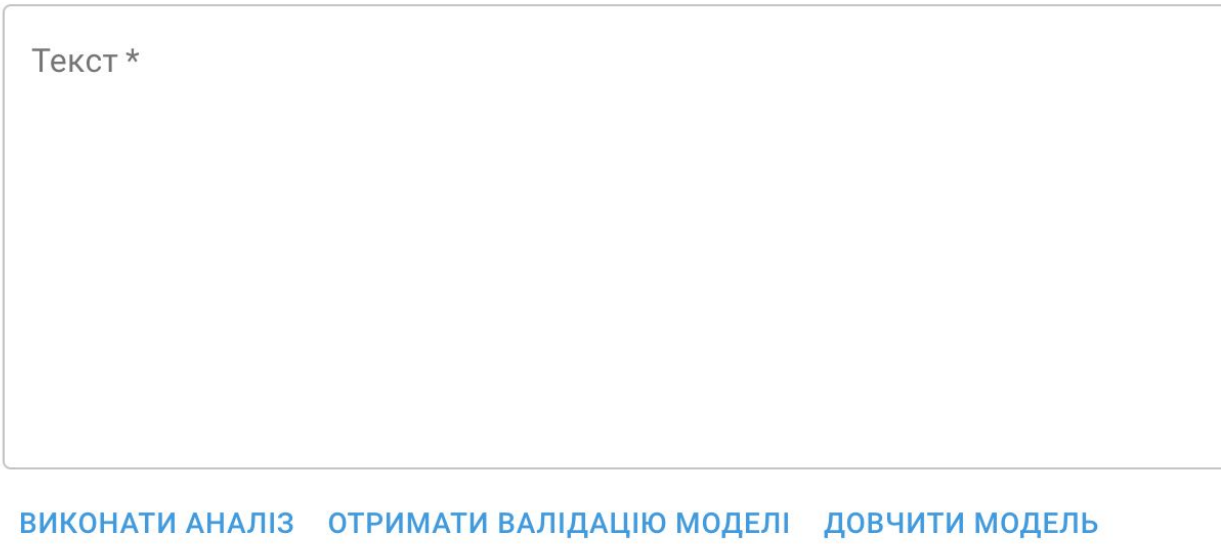


Рисунок 4.13 - Результати валідації моделі

4.3 Розгортання програмного забезпечення

Для розгортання програмного забезпечення потрібні:

- клієнт
- сервер додатку
- сервер бази даних

На сервері додатку повинно бути встановлено Node JS та Python. Версія Node JS повинна бути >= 8.0.0. Python повинен бути версії 3. Після того як встановлено Node.js потрібно зайти у командній строці у директорію проекту та

запустити команду npm install для встановлення усіх залежностей. Після встановлення залежностей запустити команду npm run start, що запустить сервер.

На сервері бази даних повинна встановлена СУБД Mongo DB. Адресу серверу бази даних потрібно ввести на сервері додатку у файлі config.js в поле mongoURL. Після цього буде встановлено з’єднання між сервером додатку та сервером бази даних.

Клієнтська частина потребує встановлення будь-якого браузера. Для використання додатку достатньо встановити один з наступних браузерів: Google Chrome, Mozilla Firefox, Internet Explorer 9+ та Safari 6+.

4.4 Висновки до розділу

У даному розділі було описано програмне забезпечення системи. Зокрема було описано сутності, що реалізовано у системі.

Також було описано організацію функціоналу додатку та основні можливості для користувача з використанням усіх трьох основних видів аналізу тексту, що було реалізовано: морфологічний аналіз, частотний аналіз та аналіз з використанням машинного навчання.

Окрім цього було написано інструкцію з розгортання додатку, зокрема розгортання клієнтської частини, серверу додатку та серверу бази даних.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

5.1 Опис ідеї проекту

Ідея проекту описується у таблицях у таблицях 5.1 – 5.22.

Таблиця 5.1 - Опис ідеї проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система для дослідження та аналізу тексту	Лінгвістичні дослідження, аналіз тексту, розпізнавання позитивної та негативної тональності тексту.	Дослідники-лінгвісти можуть проаналізувати український текст за допомогою трьох методів аналізу: лінгвістичного, статистичного та за допомогою машинного навчання. За допомогою лінгвістичного методу користувачі можуть виконати морфологічний аналіз. За допомогою частотного аналізу користувач може проаналізувати літери та біграми у тексті. За допомогою машинного навчання є можливість розпізнати тональність тексту.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Запропонований проект	Monkey Learn	Langsoft			

Продовження таблиці 5.2

1	Аналіз українського тексту	+	-	-	-	-	+
2	Наявність лінгвістичних методів аналізу	+	-	+	-	+	-
3	Наявність статистичних методів аналізу	+	-	-	-	+	-
4	Наявність методів аналізу з використанням машинного навчання	+	+	+	-	-	+

5.2 Технологічний аудит ідеї проекту

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Морфологічний аналіз	Словник морфологічних ознак слова	Розробити	Розроблено автором проекту
2	Статистичний аналіз	Частотний аналіз літер та біграм	Наявна	Доступна
3	Аналіз тональності тексту	Навчити модель за допомогою наївного байєсовського класифікатора	Розробити	Розроблено автором проекту

Продовження таблиці 5.3

4	Веб додаток	Javascript	Наявна	Доступна
	Обрана технологія реалізації ідеї проекту: для реалізації проекту було обрано мову програмування Javascript, наївний байєсівський класифікатор та методи навчання з учителем			

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

п/ п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	Велика, немає точної кількості
2	Загальний обсяг продаж, грн/ум.од	Немає точної публічної статистики
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень 4 для входу (вказати характер обмежень)	Звичка використовувати застарілі методи дослідження
5	Специфічні вимоги до стандартизації та 5 сертифікації	-
6	Середня норма рентабельності в галузі (або по 1 ринку), %	10-15%

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

п/ п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару

Продовження таблиці 5.5

	Потреба аналізувати український текст	Дослідники-лінгвісти	Загальна ком'ютерна грамотність, вік, застарілі методи	Наявність аналізу українського тексту Можливість аналізувати текст за допомогою різних методів аналізу Зручність використання та встановлення

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Велика корпорація реалізує сервіс для аналізу тексту, в якому буде наявна українська мова	Замість розробленої системи користувачі будуть використовувати систему більш відомої компанії на ринку	Залучення інвестиції до проекту, можливо об'єднання зусиль з великою корпорацією для спільної мети
2	Збільшення ціни на хостинг в постачальника	Збільшення ціни на хостинг в постачальника	Перехід на власне апаратне устаткування або у інше хмарне середовище

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання потреби аналізу українського тексту	Величезна кількість проблем, у тому числі соціальних, бізнес-проблем та інших може бути вирішена за допомогою	Чіткий опис проблем які можуть бути вирішені за допомогою системи Розповсюдження інформації про додаток, можливо з використанням реклами для того.
2	Автоматизація методів дослідження	Існування тренду автоматизації процесів у всіх сферах, у тому числі і у дослідженні українського тексту	Збільшення долі ринку, опис переваг автоматизації

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – монополія / олігополія / монополістична / чиста	Конкурентний ринок	Потреба у постійному розширенні функціоналу, покращенню UX, аналізі ринку і даних
2. За рівнем конкурентної боротьби – локальний / національний /…	Національний	
3. За галузевою ознакою – міжгалузева / внутрішньогалузева	Міжгалузева	

Продовження таблиці 5.8

4. Конкуренція за видами товарів: товарно-родова товарно-видова між бажаннями	Товарно-видова	
5. За характером конкурентних переваг - цінова / нецінова	Нецінова	
6. За інтенсивністю - марочна/не марочна	Марочна	

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замінники
	-	MonkeyLearn TextAnalyst 2.0 TagTog NetXtract WordStat WordTabulator Aylien Langsoft AOT Ventli DiscoverText Text Analyzer AlchemyLanguage	-	Дослідники лінгвісти	Ні

Продовження таблиці 5.9

Висновки:	-	Є різні можливості аналізу тексту, але українська мова відстутня	-	Рівень чутливості до зручності інтерфейсу та якості аналізу	-

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Аналіз українського тексту	Потреба користувачів у аналізу саме українського тексту
2	Можливість обрати один з трьох типів аналізу тексту	Потреба користувачів у різних типах аналізу тексту
3	Маленька компанія	Швидка зміна стратегії

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з поданим продуктом						
			-3	-2	-1	0	+1	+2	+3
1	Аналіз українського тексту	20	X						
2	Наявність трьох типів аналізу	15			X				
3	Зручність використання	15				X			
4	Розпізнавання тональності українського тексту	15		X					

Таблиця 5.12 – SWOT- аналіз стартап-проекту

Сильні сторони: аналіз українського тексту, можливість обирати один з трьох типів аналізу	Слабкі сторони: популярність компанії
Можливості: зростання потреби аналізу українського тексту, автоматизація методів дослідження	Загрози: велика корпорація реалізує сервіс для аналізу тексту, в якому буде наявна українська мова, збільшення ціни на хостинг в постачальника

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Проведення рекламних компаній	Середня	1-3 місяця
2	Розширення стандартного функціоналу	Достатня	6 місяців

5.4 Розроблення ринкової стратегії проекту

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Люди 18-25 років	Висока	Високий	Низька	Середня
2	Люди 35-45 років	Висока	Високий	Низька	Середня
3	Старше 45 років	Середня	Середній	Низька	Складна
Які цільові групи обрано: 1, 2					

Таблиця 5.15 – Визначення базової стратегії розвитку

п/ п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1		Визначення потреб дослідників лінгвістів, їх опитування, аналіз функціоналу якого вони потребують, навчання роботі з системою, розповсюдження інформації про систему	Орієнтованість на українських дослідників Систему спроектовано таким чином, що її нескладно масшабувати	Стратегія спеціалізації

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Так	Шукати нових	Так, було проаналізовано існуючі системи для іноземних ринків та виявлено 3 основні типи аналізу, серед яких лінгвістий метод, статистичний та за допомогою машинного навчання	Стратегія заняття ніші

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Аналіз українського тексту Наявність різних методів дослідження Зручність використання та встановлення системи	Стратегія заняття ніші	Функціональність	Актуальність Масштабування Зручність

5.5 Розроблення маркетингової програми стартап-проекту

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Аналіз українського тексту	Виконується аналіз українського тексту	Відсутність систем з аналізом саме українського тексту
2	Три основні методи аналізу	Можливість виконувати лінгвістичні, статистичні методи та з використанням маниного навчання	Наявність основних типів аналізу тексту

Продовження таблиці 5.18

3	Веб додаток	Відсутність потреби встановлювати додаток	Усе що потрібно для використання системи - це встановлений веб-браузер

Таблиця 5.19 – Опис трьох рівнів моделі товар

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>І. Товар за задумом</td><td colspan="3">Бажання отримати товар не виходячи з дому, повторювана покупка товарів.</td></tr>
<tr><td rowspan="5">ІІ. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>1. Лінгвістичний аналіз
2. Статистичний аналіз
3. Аналіз з використанням методів машинного навчання
4. Кількість реклами</td><td>Нм
Нм
Нм
М</td><td>Тл/Е
Тл/Е
Тл/Е
Вр/Е</td></tr>
<tr><td colspan="3">Якість: стандарти відсутні. Проект покрито тестами, регулярно тестується тестувальником.</td></tr>
<tr><td colspan="3">Пакування: веб-додаток</td></tr>
<tr><td colspan="3">Марка: Будь-який браузер</td></tr>
<tr><td rowspan="2">ІІІ. Товар із підкріпленням</td><td colspan="3">До продажу: -</td></tr>
<tr><td colspan="3">Після продажу: -</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: прихильність користувачів до функціональності та зручності системи.</td></tr>
</table>

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари замінники	Рівень цін на товари аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Середній	Середній	Будь-який	Від безкоштовної до 30$ у місяць

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Цільові клієнти – переважно дослідники лінгвісти, що потребують покращення та автоматизацію методів дослідження українського тексту	Встановлення контактів із споживачами і підтримання їх. Формування попиту. Дослідницька робота зі збору маркетингової інформації.	Канал нульового рівня	Організація збуту власними силами

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються ся цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Цільові клієнти – переважно дослідники лінгвісти, що потребують покращення та автоматизацію методів дослідження українського тексту	Пошта, месенджери, соціальні мережі, наукові товариства та наукові інститути при університетах	Інноваційність, зручність, масштабованість	Створення відчуття зручності, технічна підримка клієнтів та залучення нових користувачів..	«Розумний аналіз українського тексту»

5.6 Висновки до розділу

Основною метою даного стартап-проекту було реалізувати систему для аналізу тексту дослідниками-лінгвістами

Конкуренція у сфері досить невелика, так як під час аналізу існуючих систем було виявлено дефіцит систем для аналізу тексту українською мовою.

Таким чином реалізація стартап проекту є доцільною та рентабельною. Доцільним є також подальший розвиток проекту, так як є велика перспектива у розширенні аналізу тексту українською мовою.

ВИСНОВКИ

У даній роботі розв’язувалася задача розробки системи для дослідження та аналізу українського тексту. В роботі отримані наступні результати:

- проаналізовано існуючі системи для аналізу тексту, зокрема українського тексту. Жодна з систем не задовольняла вимоги аналізу українського тексту, що довело необхідність розробки системи. Результатом порівняльного аналізу є стаття у науковому журналі.

- проведено аналіз сучасних методів дослідження та аналізу тексту, виділено їх основні групи, серед яких лінгвістичний аналіз, статистичний аналіз та аналіз тексту за допомогою машинного навчання

- реалізовано розпізнавання тональності українського тексту за допомогою машинного навчання з використанням наївного байєсовського класифікатора та датасету, зібраного з фейсбуку. Точність розпізнавання натренованої моделі - 80.%. Також є можливість покращувати розпізнавання моделі за допомогою додавання даних до датасету.

- реалізовано частотний аналіз українського тексту, який підраховує частоту літер у тексті, а також біграм

- спроектовано архітектуру системи для аналізу тексту, яка є масштабованою, що дозволить легко додавати до неї нові методи аналізу

- розроблено веб-додаток для дослідження та аналізу українського тексту, який містить усі основі методи аналізу. Серед лінгвістичних методів реалізовано морфологічний аналіз слова, серед статистичних - частотний аналіз тексту, серед методів з використанням машинного навчання - аналіз тональності тексту

ПЕРЕЛІК ПОСИЛАНЬ

1) Monkey learn [Електронний ресурс]: (Стаття) / Sentiment analysis – Електрон. дан. (1 файл) – 2018. – Режим доступу: https://monkeylearn.com/sentiment-analysis - Назва з екрана
2) AOT [Електронний ресурс]: (Стаття) / Технологии автоматической обработки текста – Електрон. дан. (1 файл) – 2015. – Режим доступу: http://www.aot.ru/technology.html - Назва з екрана
3) Hacker Noon [Електронний ресурс]: (Стаття) / The 4 Layers of Single Page Applications You Need to Know – Електрон. дан. (1 файл) – 2018. – Режим доступу: https://hackernoon.com/architecting-single-page-applications-b842ea633c2e - Назва з екрана
4) Medium [Електронний ресурс]: (Стаття) / Patterns for designing flexible architecture in node.js – Електрон. дан. (1 файл) – 2018. – Режим доступу: https://medium.com/@domagojk/patterns-for-designing-flexible-architecture-in-node-js-cqrs-es-onion-7eb10bbefe17 - Назва з екрана
5) Щипина Л.Ю. Информационные технологии в лингвистике. Учебное пособие – 2013. С. 43-52.
6) Monkey learn [Електронний ресурс]: (Стаття) / Sentiment analysis – Електрон. дан. (1 файл) – 2018. – Режим доступу: https://monkeylearn.com/sentiment-analysis - Назва з екрана
7) SpaCy [Електронний ресурс]: (Стаття) / Everything you need to know – Електрон. дан. (1 файл) – 2017. – Режим доступу: https://spacy.io/usage/spacy-101 - Назва з екрана

8) Towards Data Science [Електронний ресурс]: (Стаття) / Top 5 Natural Language Processing Python Libraries for Data Scientist. – Електрон. дан. (1 файл) – 2019. – Режим доступу: https://towardsdatascience.com/top-5-natural-language-processing-python-libraries-for-data-scientist-32463d36feae - Назва з екрана

9) АОТ [Електронний ресурс]: (Стаття) / Технологии автоматической обработки текста – Електрон. дан. (1 файл) – 2015. – Режим доступу: http://www.aot.ru/technology.html - Назва з екрана

10) Зубова И.И. Информационные технологии в лингвистике: Учебное пособие. – МГЛУ. – Мн., 2001.

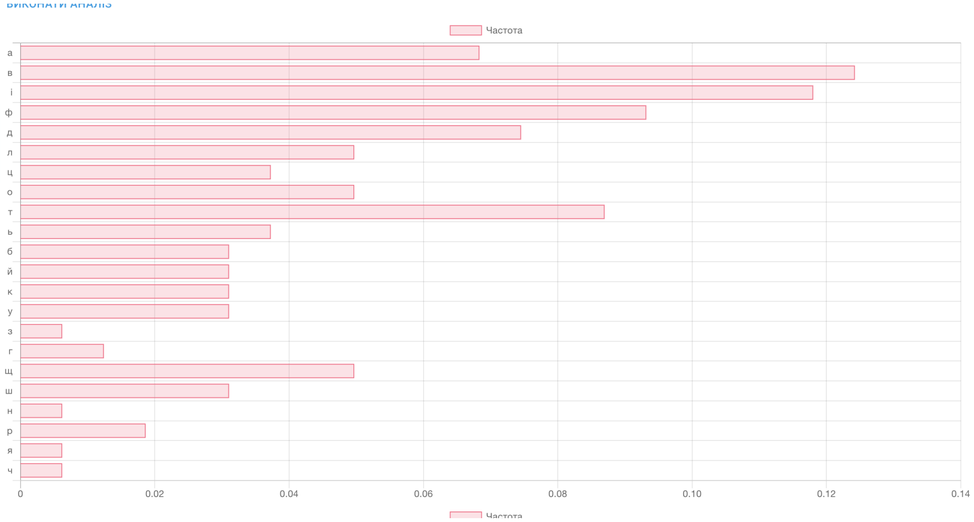
11) Русская виртуальная библиотека [Електронний ресурс]: (Стаття) / Программы анализа и лингвистической обработки текстов – Електрон. дан. (1 файл) – 2016. – Режим доступу: https://rvb.ru/soft/catalogue/c01.html - Назва з екрана

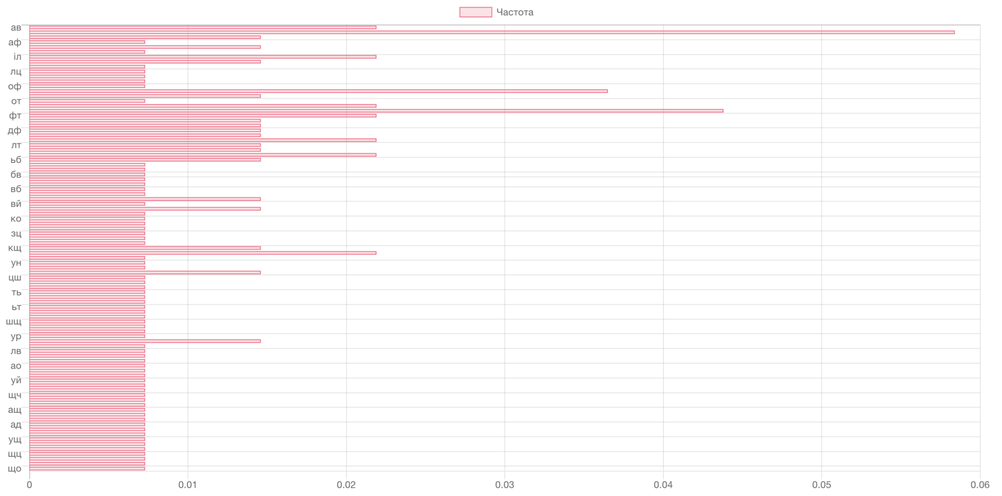
12) Семантическая поисковая система AskNet [Електронний ресурс]: (Стаття) / Программы лингвистического анализа и обработки текста – Електрон. дан. (1 файл) – 2017. – Режим доступу: http://asknet.ru/analytics/programms.htm - Назва з екрана

13) Towards Data Science [Електронний ресурс]: (Стаття) / Introduction to Natural Language Processing for Text – Електрон. дан. (1 файл) – 2018. – Режим доступу: https://towardsdatascience.com/introduction-to-natural-language-processing-for-text-df845750fb63 - Назва з екрана

ДОДАТОК А ДІАГРАМА СУТНОСТЕЙ СИСТЕМИ

ДОДАТОК Б ЕКРАННІ ФОРМИ З РЕЗУЛЬТАТАМИ АНАЛІЗУ





$$score = 100 * \frac{passed}{passed + failed}$$

70% для тренування

30% для тестування

Результати тестування:

Відсоток вгадування: 80.7034%

Загально текстів протестовано: 1281

К-ть текстів що позитивно пройшли тестування: 1021

К-ть текстів що негативно пройшли тестування: 260

	Позитивна вихідна	Негативна вихідна
Позитивна передбачена	323	237
Негативна передбачена	23	698

[Not supported by viewer]

[Not supported by viewer]

[Not supported by viewer]

[Not supported by viewer]